DECnet–RSX

Postinstallation Checkout Procedures

Order No. AA-J319C-TK

November 1983

The *DECnet-RSX Postinstallation Checkout Procedures* manual defines procedures for verifying that your installed node is operational, that connections have been established to all logically adjacent nodes, and that the communications hardware used by your DECnet-RSX node is functional.

SUPERSESSION/UPDATE INFORMATION:	This is a revised manual.
OPERATING SYSTEM AND VERSION:	RSX-11M V4.1 RSX-11M-PLUS V2.1 RSX-11S V4.1
SOFTWARE VERSION:	DECnet-11M V4.0 DECnet-11M-PLUS V2.0 DECnet-11S V4.0

To order additional copies of this document, contact your local Digital Equipment Corporation Sales Office.

digital equipment corporation · maynard, massachusetts

First Printing, April 1982
Revised, November 1983

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

Distributed and Mid-Range Systems Publications typeset this manual using DIGITAL's TMS-11 Text Management System.

MCTPEALL

Contents

Chapter 3 Node Configuration Guidelines

Chapter 4 Testing the RSX-11 PSI Installation

Appendix A Line/Circuit Testing

Appendix B DTS/DTR Test Programs

Figures

Tables

Preface

The *DECnet-RSX Postinstallation Checkout Procedures* manual identifies procedures for generating and installing a DECnet-RSX system. (Therefore, you should skim this manual prior to using the *DECnet-RSX Network Generation and Installation Guide*.) This manual also defines procedures to be observed after installation. Run these procedures to assure that:

- The system is configured and installed in accordance with defined procedures.
- The communications hardware to be used by the DECnet-RSX system is functional.
- All adjacent nodes are reachable.
- The connection between the local data terminal equipment (DTE) and the data communications equipment (DCE) is operational for DECnet-RSX systems with a packet-switching network (PSN) capability.

Intended Audience

This manual is intended for users familiar with the operation of DECnet-11M/S Version 4.0 and DECnet-11M-PLUS Version 2.0. The user should also be trained in the operation of adjacent network nodes. If your DECnet-RSX system interfaces with a PSN, you should be familiar with the operation of the PSN and the X.25 protocol.

Structure of this Manual

- Chapter 1 reviews DECnet-RSX generation and installation procedures.
- Chapter 2 describes a procedure for checking your local DECnet node and remote nodes.
- Chapter 3 provides a list of DECnet node configuration guidelines relating to routing.
- Chapter 4 describes a procedure for checking your local DTE for DECnet-RSX systems with a PSN capability.
- Appendix A describes loopback testing for lines/circuits.
- Appendix B summarizes the DECnet test sender/DECnet test receiver (DTS/DTR) command syntax.

Associated Documents

Before reading this manual, you should have a working knowledge of DECnet and the RSX-11 operating system you are using. A prerequisite to the effective use of this manual is familiarity with the overall character of DECnet as described in the *Introduction to DECnet*.

Programming and user utility information (including X.25 and X.29 programming facilities) concerning DECnet-RSX is contained in the following manuals:

DECnet-RSX Guide to User Utilities
DECnet-RSX Programmer's Reference Manual
RSX-11 PSI User's Guide
RSX-11 Utilities Manual

Network generation and system generation are described in the following three manuals:

DECnet-RSX Network Generation and Installation Guide
RSX-11M System Generation and Management Guide
RSX-11M-PLUS System Generation and Management Guide

Information specific to using the PSN capability of DECnet-RSX for a specific subscription service is contained in the *RSX-11 PSI Network-Specific Information* manual. This manual discusses various public network services.

Acronyms

CDA	Crash Dump Analyzer
CFE	Configuration File Editor
DAP	Data Access Protocol
DCE	Data communications equipment
DDCMP	Digital Data Communications Message Protocol
DLM	Data link mapping
DLX	Direct Line Access Controller
DNA	Digital Network Architecture
DTE	Data terminal equipment
DTR	DECnet test receiver
DTS	DECnet test sender
ECL	End Communication Layer
FAL	File Access Listener
FTQ	File transfer queue manager
FTS	File transfer service
HLD	Host Task Loader
LSN	Listen utility
MOP	Maintenance Operation Protocol
NCP	Network Control Program
NDA	Network Crash Dump Analyzer
NFT	Network File Transfer utility
NICE	Network Information and Control Exchange protocol
NSP	Network Services Protocol
NTD	Network Display Program
NTL	Network loader
PAD	Packet assembly/disassembly
PSN	Packet-switching network
PSI	Packetnet System Interface
RCP	Routing Control Process

SCPXTS	X.25 test sender
SLD	Satellite task loader
TLK	Terminal Communications utility
TRI	Trace interpreter task
VNP	Virtual network processor
XTR	X.25 test receiver

Graphic Conventions

SET EXECUTOR STATE OFF ON SHUT	Command example verbs and keywords are shown in a command line in uppercase letters, and they must be entered as shown. Keywords or arguments within braces indicate that you must choose only one of the keywords or arguments. (Do not include the braces when entering the command.)
SET KNOWN LINES ALL SET KNO LIN ALL	You can abbreviate any command verb or keyword to its first three characters. (This manual, however, uses the complete verb and keyword(s) in all command formats and examples.)
SET CIRCUIT *circuit-id* COST *cost*	Arguments are shown in command lines as lowercase italic letters. (In this case, you substitute the argument shown in the command format with the precise information requested.)
SET LINE XYZ . . . LOAD NODE ABC...	The use of ellipsis means that not all the information the system would display is shown. This information could be a system message, a display in response to a command, or user-entered information.
NCP>EXIT	Command examples show all output lines or prompting characters that the system displays in black letters. This manual uses red lettering to indicate all user-entered commands.
CTRL x	The expression CTRL x refers to a control character keying sequence. The key labeled CTRL and the appropriate character key should be pressed simultaneously.
RET	A symbol with a 1- to 3-character abbreviation indicates that you press a key on the terminal.
[*opt-arg*]	Square brackets indicate that the enclosed item is optional.

Chapter 1
Installing a DECnet–RSX System

This chapter outlines the steps you should have taken prior to running the postinstallation checkout procedures. Steps are grouped according to sequence of performance:

- Prior to NETGEN
- During NETGEN
- Prior to network installation
- During network installation

Only after you have performed all the steps specified should you use the checkout procedures described in Chapters 2 and 4.

You should be familiar with the information contained in the following documents:

- *DECnet-RSX System Manager's Guide*
- *DECnet-RSX Network Generation and Installation Guide*
- *DECnet-RSX Release Notes*
- *RSX-11 PSI Release Notes* (if your DECnet-RSX system interfaces with a Packet-switching Network (PSN))

1.1 Steps Prior to NETGEN

If you have not generated an RSX operating system tailored for DECnet, refer to the *DECnet-RSX Network Generation and Installation Guide*. The following sections outline additional steps you should have taken in preparation for NETGEN.

1.1.1 User Requirements

Before generating DECnet-RSX, make sure the following conditions are satisfied:

- The prerequisite software (the operating system) has been correctly generated and includes support for networks.
- The CSR and vector addresses are known to be correct for all communications hardware to be used with DECnet-RSX.
- All hardware not supplied by Digital (for example, modems and telephone lines) have been verified by the supplier to be operational.

1.1.2 Specification of Features During SYSGEN

Requirements regarding specification of functions and features during SYSGEN fall into three categories:

- SYSGEN requirements for DECnet-RSX support
- SYSGEN requirements for DECnet-RSX options desired at NETGEN
- Options, if desired, must be specified identically in both SYSGEN and NETGEN

The requirements and options are listed in the *DECnet-RSX Network Generation and Installation Guide*. The RSXMC.MAC file is an output of SYSGEN. This file contains definitions of the symbols that identify the options selected at SYSGEN. Always check the RSXMC.MAC file to verify that all required symbols are defined. The file can be checked by listing it on a hardcopy device or by displaying it on a terminal. If the required symbols are not defined, the RSX operating system must be regenerated.

1.1.3 Patches

Collect all patches that apply to your kit. Then, according to the instructions, apply the patches to a *copy* of the kit. Refer to the current *DECnet-RSX Release Notes* and *RSX-11 PSI Release Notes* before applying the patch.

1.2 Steps During NETGEN

Execute NETGEN (preferably from a hard-copy terminal) according to the procedures described in the *DECnet-RSX Network Generation and Installation Guide*. When the procedure has run to completion, review the output for errors. If there are any errors, correct them before proceeding. When you have built a system you believe will run, proceed to Section 1.3.

1.3 Steps Prior to Network Installation

To verify that you specified DECnet-RSX system information correctly during the NETGEN process, use one of the following procedures:

- Examine the network component parameters using the Configuration File Editor (CFE)
- List and examine NETCFE.CMD and NETCFG.TXT, both located under the network UIC [*x,x,x*,1]

These techniques are described in the following sections.

1.3.1 Use of the Configuration File Editor (CFE)

CFE can be used to examine the data for the network configuration. (For more information on CFE, refer to the *DECnet-RSX System Manager's Guide*.) Use the following CFE commands to determine how the component parameters were set during NETGEN:

LIST SYSTEM

LIST EXECUTOR

LIST KNOWN LINES

LIST KNOWN CIRCUITS

LIST KNOWN LOGGING

LIST KNOWN NODES

LIST KNOWN OBJECTS

The following CFE commands are useful only in DECnet-RSX networks that interface with a packet-switching network by means of the Packetnet System Interface (PSI) product:

LIST MODULE X25-ACCESS

LIST MODULE X25-PROTOCOL

LIST MODULE X25-SERVER

LIST MODULE X29-SERVER

1.3.2 Examination of NETCFE.CMD and NETCFG.TXT

The NETCFE.CMD file is a journal of all the commands entered and executed during NETGEN. This file also lists the CFE commands executed in the selection of any of the default values implicitly or explicitly specified during NETGEN.

The NETCFG.TXT file contains a narrative description of the system configuration as formed during NETGEN.

Both files are a product of network generation. They accurately reflect the configuration of the network at NETGEN. To use the files' information, list one or both on a hard-copy device and verify that the results of the network generation reflect the options that you intended to specify.

1.4 Steps During Network Installation

Once the network parameters have been verified, the software can be installed. Refer to the *DECnet-RSX Network Generation and Installation Guide* and the *DECnet-RSX System Manager's Guide* for guidelines and instructions for bringing up your DECnet-RSX system.

After the node is up on the target system, its operation should be verified as follows:

- If the system includes DECnet support over Ethernet, leased, or private lines, follow the procedures described in Chapter 2.
- If the system includes the capability to interface with a PSN, test that capability by following the procedures described in Sections 4.1 through 4.3.
- If the system includes DECnet support over PSN facilities (data link mapping), follow the procedures given in Chapter 4 to check the operation of DLM circuits.

Chapter 2
Installation Testing

This chapter describes procedures you can use to check your RSX-11S, RSX-11M, or RSX-11M-PLUS node. Certain restrictions, as outlined in Section 2.4, apply to testing an RSX-11S node. The checkout procedures consist of three types of tests:

- The first test verifies that your local node software is operational.
- The second test verifies that your local DECnet node and communications hardware are functioning properly.
- The third test checks your local node's ability to communicate with a remote node (see Figure 2-1).

All procedures are designed to be initiated and executed from the local node. However, an operator should be available at each remote node to assist in problem isolation. To facilitate problem isolation for remote loopback testing, you should enable the event logger if you have not done so already or if event logging has been turned off with CFE or NCP. (If event logging was selected as an option, the default state is ON as a result of NETGEN.) You can enable event logging by using the following command:

```
NCP SET LOGGING CONSOLE STATE ON
```

For more information on the Network Control Program (NCP) commands mentioned in this chapter, refer to the *DECnet-RSX System Manager's Guide*.

2.1 Local Node Software Testing

This test verifies that five layers of software are working: the User layer, the Network Management layer, the Network Application layer, the End Communications layer, and part of the Routing layer. This test does not check the device driver or the communications device. See Figure 2-2 for the path the test data travels.

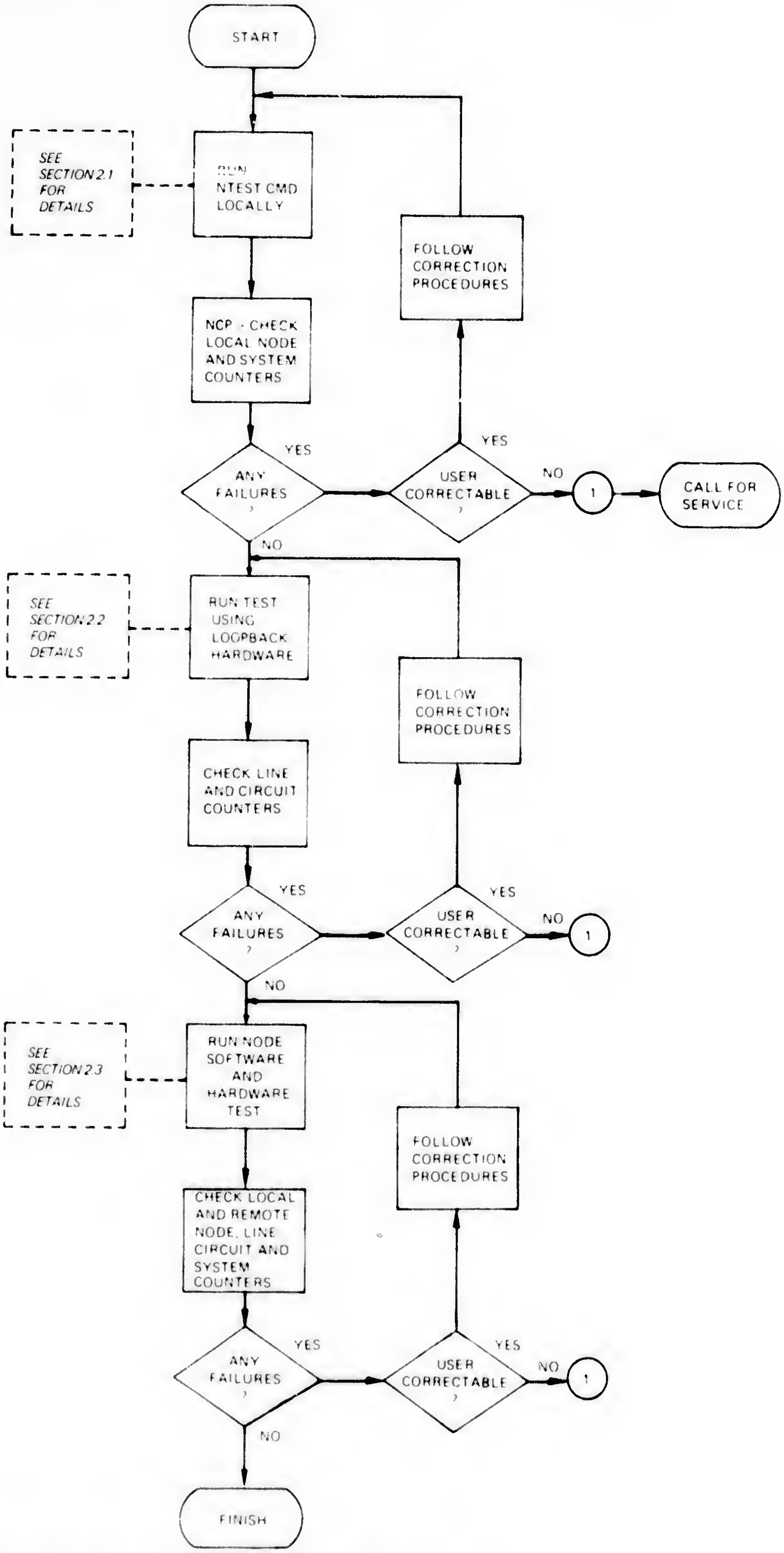


Figure 2-1: Installation Test Flowchart

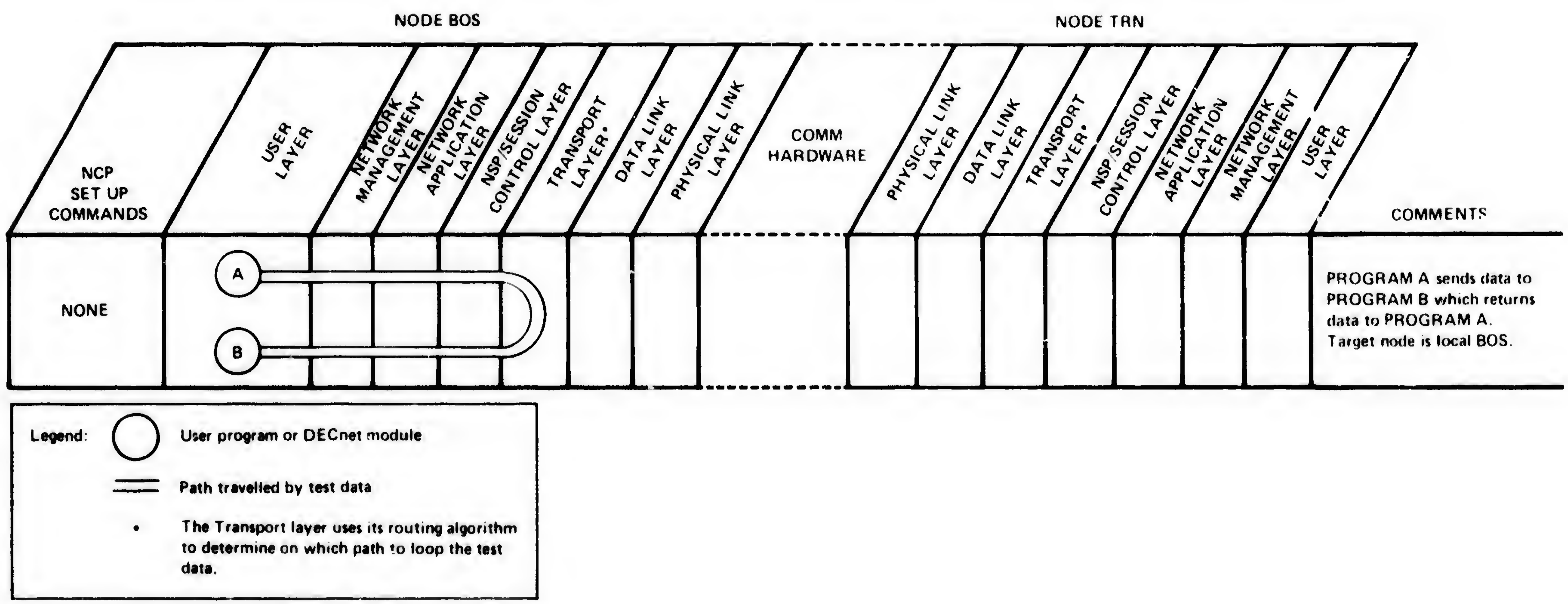


Figure 2–2: Local Node Software Test

You may use other types of tests, but the NTEST.CMD procedure is preferred because it is automated. The NTEST.CMD procedure tests and installs the network tasks needed for testing and brings up the network.

If the requirements of Section 2.1.1 cannot be met (for example, a 16K-word system controlled partition), the TLK and LSN utilities can be substituted for the NTEST.CMD test.

NTEST.CMD executes two sets of programs: DECnet test sender (DTS) and DECnet test receiver (DTR), and Network File Transfer (NFT) and File Access Listener (FAL). DTS/DTR perform message transfer operations using task-to-task communication. NFT/FAL perform various file transfer operations. Details of the DTS/DTR programs are in Appendix B. Details of NFT and FAL are found in the *DECnet-RSX Guide to User Utilities*.

2.1.1 Preparing to Run the Test

The following conditions must be met before the local NTEST.CMD procedure is executed:

- The partition in which the network software and the software tools will run must be system controlled and at least 16K words in length.
- A NETGEN must be completed.
- NETGEN must have moved the NTEST.CMD file onto your disk under the UIC [*xxx*,24], where *xxx* is the group code chosen during NETGEN.
- You must be logged into a privileged account (if your system supports multiuser protection).

When you have satisfied the conditions listed above, proceed to the next section and run NTEST.CMD.

2.1.2 Running the Test

First, set the network UIC number under which you stored NTEST.CMD. You can set this number using the following command:

```
SET /UIC=[xxx,24] RET
```

where *xxx* is the network software UIC group code specified during NETGEN.

Next, invoke the command file by entering the following:

```
@NTEST.CMD RET
```

NTEST.CMD begins to execute and prompts you for any input it requires during execution. Figure 2-3 is a sample listing of the NTEST.CMD procedure.

NOTE

> Your answers to the questions displayed by the NTEST.CMD procedure may be different from those shown in the sample listing. For example, you may specify device names, YES and NO options, and group UIC codes that differ from those shown.

```
@NTEST
SET /BUF=TI:80
;
;=============================================================
;               RSX-11M/M-PLUS Network Installation Procedures
;=============================================================
;
; Copyright (C) 1979, 1980, 1981, 1982, 1983 by
; Digital Equipment Corporation, Maynard, Mass.
;
SET /UIC=[1,1]
; Test procedure performed at 08:02:34 on 01-AUG-83
;
; This procedure is designed to allow a user to perform the following
; tests on a DECnet-11M/M-PLUS system:
;
;   .  Internal node level -- Test the local node without using any
;      communication circuits.
;   .  Circuits level -- Test the local node with a communications line
;      attached to a turnaround/loopback device or a modem with loopback
;      capabilities.
;   .  Remote node level -- Test the local node by attempting to communi-
;      cate with a remote node.
;
;
; The procedure will execute DTS/DTR tests and will use NFT/FAL to
; transfer and execute files.
;
; Before running this procedure, you should have access to the following
; information:
;
;   .  The device and group code of the UIC under which the network
;      tasks are stored.
;   .  The target node name (for local tests, this is the local node
;      name).
;   .  The necessary UIC and password to access files residing on the
;      target node, for NFT tests.
;
 Do you want to: CR-continue E-exit P-pause [S]:
;
;=============================================================
;                            Test Parameters
;=============================================================
;
; If you do not choose long dialogue mode, explanatory text for each
; question is available by hitting the "ESCAPE" key.
;
 Do you want long dialogue mode? [Y/N]:
;
; Enter the UIC group code under which the network files were stored
; for this node's network.
;
 What group UIC is the network stored under [O R:1-377]:
;
; Specify the device on which the network files are stored. The default
; device is "SY:".
;
 On what device are the network files (DDNN) [S R:0-5]:
ASN DM1:=SY:
;
; Answer "YES" to the next question if this is an internal node test,
; i.e. the test does not involve communications lines and/or
; remote nodes.
;
 Is this an internal node test? [Y/N]:
```

(continued on next page)

Figure 2-3: NTEST.CMD Sample Listing

```
;
; Enter the target node name (1-6 alphanumeric characters). This is the
; node name of the remote node. If this is an internal level test, then
; the local node name should be used. If this is a line level test, then
; the loopback node name should be used.
;
What is the local node name [S R:1-6]: [illegible]
;
; DTS will now be invoked to perform three tests:
;        1.)  DTS will issue a connect request to DTR causing
;             DTR to be loaded, answer the request and then
;             exit
;        2.)  DTS/DTR will execute a 1 minute pattern test
;             using a 10 byte message size
;        3.)  DTS/DTR will execute a 1 minute pattern test
;             using an 1100 byte message size.
;
; The 10 byte message size was selected because this size is smaller
; than ECL's segment size...thus there will be 1 segment for each
; message. The 1100 byte message size is larger than the segment size;
; therefore, there will be multiple segments for each of these messages.
;
REM ...DTS
REM DTR...
INS DM1:[[illegible],54]DTS/INC=[illegible]
INS DM1:[[illegible],54]DTR/INC=[illegible]
PIP DTSTST.CMD;* DE/NM
DTS @DTSTST
LOAD & EXEC DTR VIA CONNECT TEST
DTS -- Test finished at [illegible]
PAT MSG=10 TIME=1M
DTS -- Test started at [illegible]
DTS -- Test finished at [illegible]
[illegible] Messages sent
[illegible] Characters/second
[illegible] Baud

PAT MSG=1100 TIME=1M
DTS -- Test started at 8:04:[illegible]
DTS -- Test finished at 8:05:[illegible]
876 Messages sent
16063 Characters/second
128490 Baud

PIP DTSTST.CMD;* DE/NM
;
;
;
; In order to execute NFT file transfer tests, you must supply valid
; access control information for the target node NODEA. This
; information is
;        (1). A user identification (the format depends on what type of
;             system NODEA is).
;        (2). The password for that user id.
;
What is the userid for NFT tests on node NODEA [S R:[illegible]]:
What is the corresponding password [S R:[illegible]]:
PIP NFTTST.ZZZ;* DE/NM,[illegible]
;
; NFT will now be invoked. A copy of the command file NTEST.CMD will be
; transferred to node NODEA with the filename TMPNETTST.ZZZ using the
; access control information you specified. NFT will then delete the
; file.
```

(continued on next page)

Figure 2-3 (cont.): NTEST.CMD Sample Listing

```
 :
 NFT @NFTTS1.ZZZ TR
NFT NODEA [200,200] PRIV::TMPNETTST.ZZZ;1=DM1:[100,24]NTFS1.CMD
 :
 : The file TMPNETTST.ZZZ will now be deleted using NFT.
 :
 NFT @NFTTS2.ZZZ TR
NFT NODEA [200,200] PRIV::TMPNETTST.ZZZ;1 DE
 :
 PIP NFTTS1.ZZZ; DE NM,NFTTS2;,NFTTS3;,NFTTS4;,NFTTS5;
 @SN:
 :
 : If you have reached this point with no errors, then you can be
 : confident that the node has been created properly.
 :
 : End of test at 08:07:12
 :
 :=================================================================
 :                           End of Installation Test
 :=================================================================
 SET  UIC=[100,24]
 @  EOF
```

Figure 2-3 (cont.): NTEST.CMD Sample Listing

Now, use the **NCP SHOW EXECUTOR COUNTERS** and **SHOW SYSTEM COUNTERS** commands to display the counters. Check the counters for errors. If there are any errors, proceed to Section 2.1.3.

If there are no errors, proceed to Section 2.2, Node Level Communications Circuit Testing, using a hardware loopback device.

The following counters reflect an error-free test execution. Upon entry of the **NCP SHOW EXECUTOR COUNTERS** command, the following is displayed at your terminal:

```
Node counters as of 01-MAR-83 09:24:12

Executor node = 14 (NODEA)

          308  Seconds since last zeroed
      1009182  Bytes received
      1009182  Bytes sent
        17658  Messages received
        17658  Messages sent
            8  Connects received
            8  Connects sent
            0  Response timeouts
            0  Received connect resource errors
            2  Total maximum logical links active
            0  Aged packet loss
            0  Node unreachable packet loss
            0  Node out-of-range packet loss
            0  Oversized packet loss
            0  Packet format error
               [illegible]
            0  Verification reject
            2  Node maximum logical links active
            0  Total received connect resource errors
```

To display the system counters, enter

```
NCP>SHOW SYSTEM COUNTERS
```

The following is displayed at your terminal:

```
System counters as of 01-MAR-83 09:25:03

          368  Seconds since last zeroed
            0  Control buffer allocation failed
            0  Small buffer allocation failed
            0  Large buffer allocation failed
            0  Receive buffer allocation failed
```

2.1.3 NTEST.CMD Failure Conditions and Handling

The most common failure conditions encountered during this test are resource allocation failures or network tasks not installed. The symptoms most commonly seen under these failure conditions are described below. Identify which class of failure is occurring, take appropriate corrective action, and then repeat the test.

Common symptoms of this class of failure are:

- Inability to execute the NCP SET SYSTEM command
- Inability to execute the NCP SET EXECUTOR STATE ON command
- Inability to establish a logical link

Corrective action. If, at any time, the operation appears to be hung, use the NCP SHOW SYSTEM COUNTERS command. If any of the counters are incrementing, there is a resource allocation problem. The network should be unloaded, and the resource allocations should be increased using CFE. Once the system resources have been increased, reload the network with the NCP SET SYSTEM command. Then set the executor state ON with the NCP SET EXECUTOR STATE ON command.

Allocation failures. Resource allocation failures include one or more of the following conditions:

- Insufficient number of large buffers (LDBs)
- Insufficient number of control buffers (CCBs)
- Insufficient number of small buffers (SDBs)
- Insufficient RSX system pool size (DSR)
- Insufficient number of logical links

Other failures. Other reasons for failure can include one or more of the following conditions:

- Insufficient checkpoint space
- Fragmented system-controlled partition
- Hardware problems, including disk errors that can result in nonreproducible system behavior

If there is insufficient checkpoint space, RCP1 (routing control process) may not be able to allocate space for its data base and you would receive the following message:

```
RCP1 - Cannot allocate space for data base
```

For more information on network load failures, refer to the *DECnet-RSX Network Generation and Installation Guide.*

If NTEST.CMD fails while running DTS/DTR or NFT/FAL, refer to Appendix B (for DTS/DTR) or to the *DECnet-RSX Guide to User Utilities* (for NFT/FAL).

2.2 Node Level Hardware Loopback Circuit Testing

This test verifies that the Routing layer, the device drivers, and the communications hardware are functioning properly. In addition to the layers tested in the local node software tests (Section 2.1), the Data Link and Physical Link layers and associated hardware are tested in this procedure.

This test transfers data over the communications circuit through the loopback connector to the associated loop node name. Figure 2-4 shows the path the test data travels. As illustrated in Figures 2-6 and 2-7, the test can be run to the controller, to the local modem, and to the remote modem.

Observe these specific instructions for the type of device you are testing.

- For UNA and QNA devices, proceed to Section 2.3.
- For PCL, PCL-*n*.0 is assigned as the local TDM address. This circuit should be used for the tests specified in this section.
- For DLM, run the PSI checkout procedures first, including NCP LOOP LINE; then proceed to Section 2.3.
- For multipoint devices (controller or tributary) using software DDCMP, proceed to Section 2.3.
- For half duplex devices (point to point) temporarily set up to full duplex for the hardware local modem loopback test, if successful, switch back to half duplex and proceed to Section 2.3.
- For all other devices, proceed to Section 2.2.1.

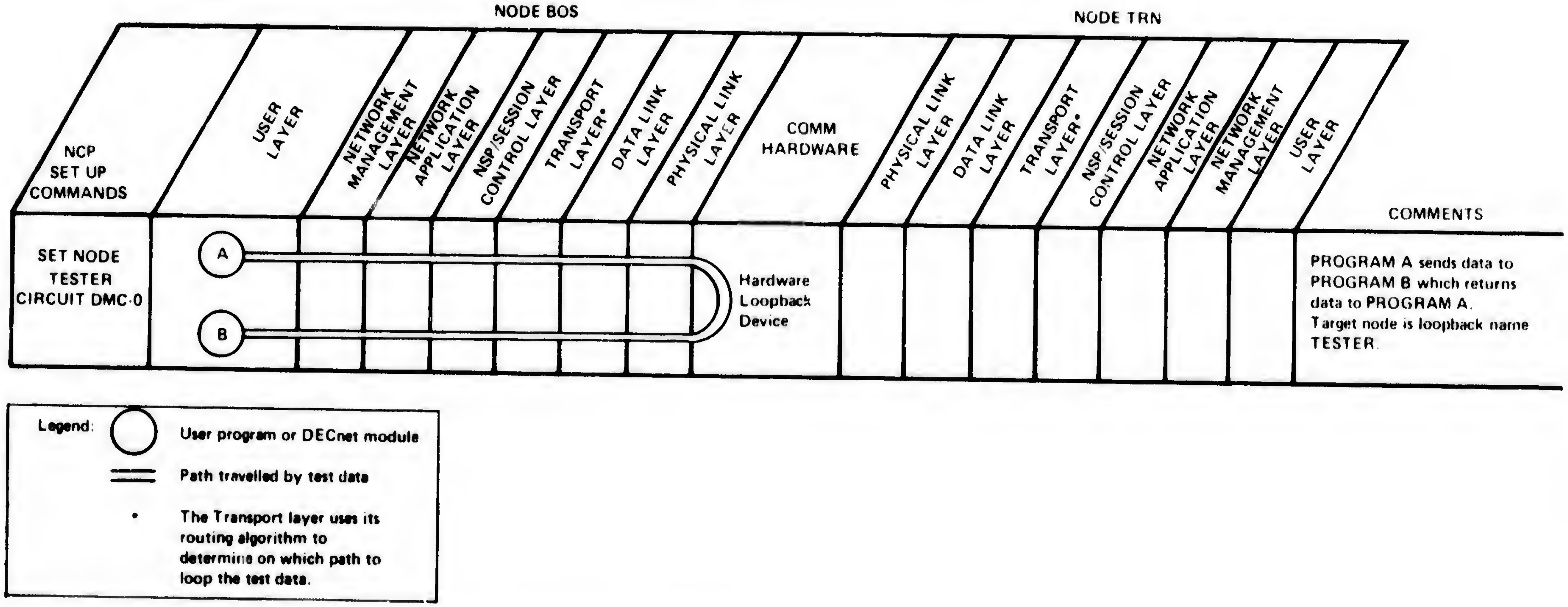


Figure 2-4: Node Level Test Using a Loopback Connector

2.2.1 Preparing to Run the Test

Perform three steps before running the test:

Step 1:

- Install the hardware loopback connector using the connector listed in Table A-1
- Use the analog loopback button on the modem (see Section A.1.1)
- Set the device (DMC/DMR, DMP, or DMV) to controller loopback with the following NCP commands:

```
NCP [illegible]
NCP [illegible]
```

Step 2:

Use the following NCP commands to set the line and circuit on (assuming that the line is not already loaded). For example, enter

```
NCP [illegible]
NCP [illegible]
```

The event logger prints a message on the console if logging is enabled. A sample message is shown below.

```
Event type 4.10, Circuit up
Occurred 01-MAR-83 09:29:17 on node 14 (NODEA)
Circuit DMC-0
Node address = 14 (NODEA)
```

Step 3:

Use the NCP SHOW command to verify that the circuit state has changed to ON. Note that the executor node is now also the adjacent node on the test circuit.

```
NCP [illegible]
```

A sample of the summary displayed is shown below:

```
Circuit summary as of 01-MAR-83 09:42:56

Circuit = DMC-0

     State = On
     Adjacent node = 14 (NODEA)
```

If the state on the circuit is still on-starting, proceed to Section 2.2.5 (On-Starting Substate Error Handling).

2.2.2 Running the Loop Node Test

Use the SET NODE command with the CIRCUIT parameter to establish a loop node name. For example,

```
NCP>[illegible]
```

establishes circuit DMC-0 as the circuit over which loop testing takes place. The loop node name is necessary because, under normal operation, DECnet routing software decides what path to use. The loop node name overrides the routing function to route information over a specific circuit.

Next, use the NCP command to loop messages over the line through the loopback connector to the loopback node name TESTER. (For a more extensive test, you can run the NTEST.CMD procedure described in Section 2.1.2. When NTEST asks you if it is an internal test, answer NO.)

```
NCP>[illegible]
NCP>[illegible]
```

A sample of the summary displayed is shown below:

```
Circuit counters as of 01-MAR-83 09:54:15

Circuit = DMC-0

        220   Seconds since last zeroed
         87   Terminating packets received
         87   Originating packets sent
          0   Transit packets received
          0   Transit packets sent
          0   Transit congestion loss
          0   Circuit down
          0   Initialization failure
       4378   Bytes received
       4378   Bytes sent
        143   Data blocks received
        143   Data blocks sent
          0   Data errors inbound
          0   Data errors outbound
          0   Remote reply timeouts
          0   Local reply timeouts
          0   Local buffer errors
```

If any of the last five counters shows a nonzero value, see Section 2.2.6; otherwise enter the following command to show system counters:

```
NCP>[illegible]
```

A sample of the summary displayed is shown below:

```
System counters as of 01-MAR-83 09:56:42

       1860   Seconds since last zeroed
          0   Control buffer allocation failed
          0   Small buffer allocation failed
          0   Large buffer allocation failed
          0   Receive buffer allocation failed
```

Errors indicated by the system counter summary can be corrected by increasing the number of buffers whose allocation failed. Use CFE to modify system buffer parameters.

Next, enter the command to show the executor node counters:

```
NCP [illegible]
```

A sample of the summary displayed is shown below:

```
Node counters as of 11 MAR 83 09:56:[illegible]

Executor node = 1.4 (NODEA)

       1861   Seconds since last zeroed
    1011764   Bytes received
    1011764   Bytes sent
      17745   Messages received
      17745   Messages sent
          9   Connects received
          9   Connects sent
          0   Response timeouts
          0   Received connect resource errors
          2   Total maximum logical links active
          0   Aged packet loss
          0   Node unreachable packet loss
          0   Node out-of-range packet loss
          0   Oversized packet loss
          0   Packet format error
          0   Partial routing update loss
          0   Verification reject
          2   Node maximum logical links active
          0   Total received connect resource errors
```

Check the summaries. If the counters were zeroed before the test, then the latest summary should show equivalent values for the following pairs:

Circuit Counter Pairs	**Node Counter Pairs**
Terminating packets received Originating packets sent	Bytes received Bytes sent
Transit packets received Transit packets sent	Messages received Messages sent
Bytes received Bytes sent	Connects received Connects sent
Data Blocks received Data Blocks sent	
Data errors inbound Data errors outbound	
Remote reply timeouts Local reply timeouts	

Check the following circuit counters: data errors inbound, data errors outbound, remote reply timeouts, and local reply timeouts. If no errors are found, proceed to Section 2.2.3. If errors are found, see Section 2.2.6.

2.2.3 Restoring the Node

Now that the test is complete, the circuit must be restored to its normal state. This involves removing the loopback name, removing the hardware loopback connector, or setting the line back to controller normal (from controller loopback).

Use the NCP CLEAR NODE command to remove the loopback name. The loopback name is TESTER so enter:

```
NCP>CLEAR NODE TESTER CIRCUIT
```

Next, remove the hardware loopback connector. Note that the circuit status returns to the on-starting state. The time it takes to return to the on-starting state depends on the listener time for the circuit. The default value specified in NETGEN is 30 seconds. A sample of the event message displayed is shown below:

```
Event type 4.7, Circuit down - circuit fault
Occurred 01-MAR-83 10:05:16 on node 14 (NODEA)
Circuit DMC-0
Adjacent node listener receive timeout
```

If a DMC/DMR, DMP, or DMV device was previously set to controller loopback, you must set the device back to controller normal (the default state) with the following commands:

```
NCP>SET CIRCUIT DMC-0 STATE OFF
NCP>SET LINE DMC-0 CONTROLLER NORMAL
NCP>SET CIRCUIT DMC-0 STATE ON
```

The local node level communications circuit testing is now complete. You can be confident that the local software and hardware are functioning properly and can connect to a remote node. Proceed to Section 2.3.

2.2.4 Setting Line/Circuit States

When NETINS.CMD performs the NCP SET SYSTEM command, all lines that have been previously set to ON (during NETGEN) are enabled. After NETGEN is completed, there are two ways you can enable or disable lines.

You can either change the permanent data base by using the Configuration File Editor (CFE) commands, or you can use the Network Control Program (NCP) commands to enable a line for the currently running system.

Using CFE. If you want to enable a line using CFE commands, modify the permanent data base either before you execute NETINS.CMD or before you do the next NCP SET SYSTEM. To enable a line using CFE, enter

```
CFE> DEFINE LINE line-id STATE ON
```

If you want to disable a line that has been previously set to ON, enter

```
CFE> DEFINE LINE line-id STATE CLEARED
```

Using NCP. If you want to enable a line or circuit using NCP commands, enter the following commands after the NETINS.CMD file completes execution:

```
NCP  SET LINE line-id TOP
```

This command loads the associated process in the top of memory.

```
NCP  SET CIRCUIT circuit-id STATE ON
```

This command enables the circuit.

If you want to disable a line that has been previously set to ON, enter

```
NCP  SET CIRCUIT circuit-id STATE OFF
NCP  CLEAR LINE line-id ALL
```

2.2.5 On-Starting Substate Error Handling

The test cannot be run if the circuit state is on-starting. Examine the hardware loopback connector. Be sure it is properly installed and conforms to the specifications in Appendix A. If you are using a modem with the analog loopback capability, see if special strapping is required. See Section A.2 on type 201C modems.

Next, use the NCP SHOW LINE *line-id* CHARACTERISTICS command to check that the line has been set to full duplex mode (even though you may have selected half duplex during NETGEN). Half duplex mode may have been specified in the network generation procedure. Use NCP commands to change the controller; for example,

```
NCP SET CIRCUIT DMC-0 STATE OFF
NCP CLEAR LINE DMC-0 ALL
NCP SET LINE DMC-0 DUPLEX FULL
NCP SET LINE DMC-0 CONTROLLER LOOPBACK
NCP SET CIRCUIT DMC-0 STATE ON
```

If the circuit could not be set ON, check

- For resource allocation problems. NCP>SHOW SYSTEM COUNTERS should show no errors.
- That CSRs and vectors were properly specified during NETGEN. If they were not, correct them by means of NCP or CFE.
- If a device was included in the RSX system during SYSGEN and also included at NETGEN. If it was, the following message will appear when loading the network:

```
NTL-Vector not available
```

If this condition occurs, you need to change the hardware vector and CSR of the device, as well as modify the network permanent data base, (or re-SYSGEN). Otherwise check with field service.

See the *DECnet-RSX Network Generation and Installation Guide* for detailed SYSGEN requirements. If the circuit is still in the on-starting state, use the NCP SHOW CIRCUIT COUNTERS command. Check the summary. If bytes received and bytes sent are zero, the communications hardware is at fault. Have Digital field service verify that the device and connector have been properly installed.

Three common hardware failures that occur are:

- The communication device fails to interrupt.
- The modem and/or line corrupts the data being transmitted or received.
- Communication device addresses (CSR and vector) are improperly set on the interface.

Common symptoms of this type of failure are:

- Inability to turn on a circuit
- Inability to complete the transport initialization operation

Most other types of failures are documented by appropriate error messages. You should take advantage of the extensive status and error counters maintained by the DECnet-RSX system to aid in trouble shooting. Refer to the *DECnet-RSX System Manager's Guide* for descriptions of the various counters and procedures for displaying them.

2.2.6 Line and Circuit Counter Error Handling

The following errors indicate problems requiring Digital field service:

- Errors indicated by circuit counters with receive overrun errors qualifiers can indicate that the communication device is installed at the wrong device priority. For example, a bus request plug may not be installed at BR5.
- Data errors inbound or outbound indicate a hardware device, modem, or line problem.

2.3 Remote Node Software Testing

In this test, all the network layers are tested. Figure 2-5 depicts the path taken through the various layers.

2.3.1 Preparing to Run the Test

Prepare the system for the test by means of these steps:

1. Use the NCP SHOW KNOWN CIRCUITS command. If your circuit is in the on-starting substate, proceed to Section 2.3.3 for non-Ethernet devices and Section 2.3.4 for Ethernet devices. Follow the procedures in those sections to get the circuit state to ON. If the circuit state is ON, proceed to the following step. For Ethernet devices, you can test any adjacency (remote node on the Ethernet) that is reachable. Use the NCP SHOW NODE *nodeid* command to determine if an adjacency is reachable.
2. Clear all the counters used in previous tests by entering the NCP ZERO COUNTERS commands, including system, executor, and circuit.

NOTE

For remote tests only, both DTR and FAL must be installed on the remote node before you can run NTEST.CMD.

2.3.2 Running the Test

There are two methods of testing. One method uses the NTEST.CMD file, the other method uses DECnet utilities. You can use either method. When testing is completed, display all counters using NCP commands. Be sure to include circuit, node, executor, and system counters. Pairs of counters will no longer match on the local node, but errors should be noted and corrected. Refer to the *DECnet-RSX System Manager's Guide* for information on correcting errors indicated by the network counters.

2.3.2.1 NTEST.CMD — This test is the same as the one described in Section 2.1.2, with one exception. Instead of answering the question, Is this an internal node test, with a Y, answer with an N. Then follow the procedures described in that section.

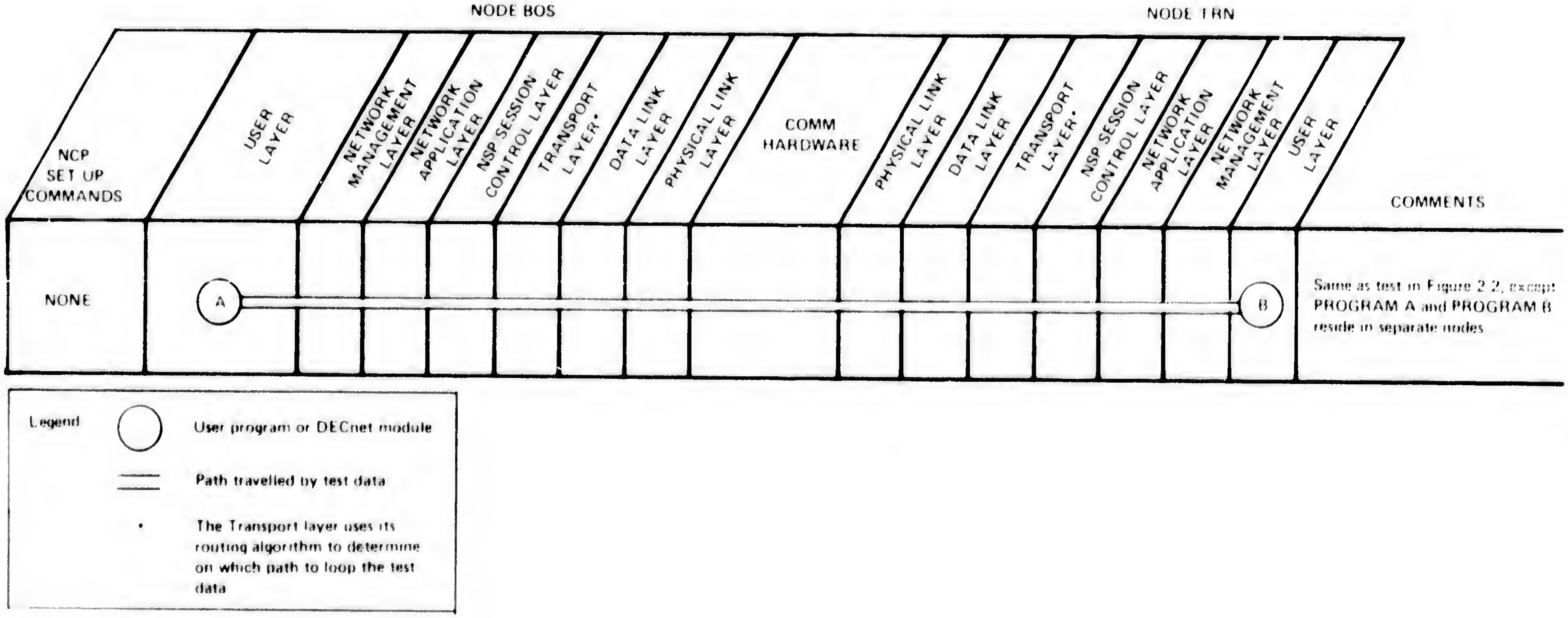


Figure 2-5: Remote Node Software Test

2.3.2.2 DECnet Utility Tests — When all circuits have been verified to be in the ON state, run one of the DECnet utilities listed in Table 2-1. Utilities are listed in descending order of preference. The DTS/DTR test is more thorough than the others, which is why it is the preferred test. The table specifies the availability of ea test on each of the DECnet products.

Table 2-1: Utility Availability on DECnet Nodes

TEST	REMOTE NODE TYPE					
	DECnet-RSX	DECnet-20	DECnet-RT	DECnet-E	DECnet-VAX	DECnet-IAS
DTS/DTR	Yes	Yes*	Yes	Yes	Yes	Yes
NFT/FAL	Yes	Yes	Yes	Yes	Yes	Yes
TLK/LSN	Yes	No	Yes	Yes	No	Yes

* Available from software specialist

DTS/DTR. In order for the DTS/DTR utility to run, DTR must be installed at the remote node. The general format for DTS is shown below, followed by an example. Full details on DTS/DTR are given in Appendix B.

Format:

DTS *nodename[aci]*::[/TEST *type*][/PRINT *choice*]

(*aci* is optional access control information.)

Example:

```
DTS REMNOD::/TEST:DATA
```

NFT/FAL. In order for NFT to perform a file operation to a remote node, FAL must be installed at the remote node.

Use NFT to transfer a copy of the LB:[1,2]STARTUP.CMD file to the remote node and then back to the local node. Then, use PIP to output the file and determine wheth the file was transmitted and received correctly. (You can use the CMP uti to compare the two versions of the file.) An example is shown below.

```
>NFT
NFT>REMNOD"10,10 PASWD"::=LB:[1,2]STARTUP.CMD
NFT>=REMNOD"10,10 PASWD"::STARTUP.CMD
^Z
>PIP TI:=STARTUP.CMD
```

Note that the file is transferred from the local node's UIC[1,2] to the remote node's UIC[10,10]. The file is then returned to the local node's UIC. Access control information is given in the example, but can be defaulted if an alias node name is specified.

For more information on NFT, refer to the *DECnet-RSX Guide to User Utilities*. There may be special file considerations, if the remote node is not RSX, as the syntax of the specified file can differ from that expected by NFT. (Check the appropriate system documentation for details.)

TLK/LSN. In order for the TLK/LSN utility to run, LSN must be installed at the remote node, and an operator must be present to respond to the message. The command syntax for single message mode is shown below, followed by an actual example.

Format:

>TLK [*targetnode*::][TT*n*:]'*message*

(TT*n*: defaults to the console device.)

Example:

```
TLK REMNOD::IF THIS MESSAGE IS RECEIVED, WE HAVE SUCCEEDED.
```

For information on the use of TLK on RSX systems, refer to the *DECnet-RSX Guide to User Utilities*. When using TLK on other systems, check the appropriate system documentation.

2.3.3 Non-Ethernet Error Handling

The circuit state must be ON in order to run remote tests. If the circuit has a substate of STARTING, the remote operator should make sure that the remote node's corresponding circuit also has been set to the ON state. If the STARTING substate persists upon reexecution of the NCP SHOW KNOWN CIRCUITS command, use the NCP SHOW EXECUTOR CHARACTERISTICS command (shown below) to see whether transmit and receive passwords have been set on the local system. If passwords have been set, make sure that the passwords are the same on the remote node. An example using the executor node follows.

```
NCP>SHOW EXECUTOR CHARACTERISTICS
```

This command displays the following information:

```
Node characteristics as of 01-JUN-83 13:34:15

Executor node = 16 (EOWYN)

Identification = EOWYN BL 19 DECNET, Management version = 4.0.0
Host = 16 (EOWYN), Loop count = 1, Loop length = 40
Loop with = Mixed, NSP version = 4.0.0
Maximum links = 10, Routing version = 2.0.0, Type = Routing IV
Routing timer = 300
Maximum address = 400, Maximum circuits = 6, Maximum cost = 1022
Maximum hops = 10, Maximum visits = 20
Maximum broadcast nonrouters = 32
Maximum broadcast routers = 10
Transmit password = ...
Verification state = On
```

If passwords have been set, they must have been set appropriately on both nodes. If the passwords are not identical, or if a password was set at one node and not at the other, the following event messages are generated at the two nodes (provided that the event logger has been enabled):

- Message at node requesting password:

```
Event Type 4.6 Verification Reject
From Node 9 (HARTFD) Occurred 05-Feb-83 17:16:21
```

- Messages at node with incorrect or missing transmit password:

```
Event Type 4.10, Circuit Up
From Node 4 (BOSTON) Occurred 05-Feb-83 17:16:55
Circuit DMC-1

Event Type 4.7 Circuit Down, Circuit Fault
From Node  4 (BOSTON) Occurred 05-Feb-83 17:17:12
Circuit DMC-1
```

If a requesting node does not receive a password or receives an incorrect one, it restarts the node initialization process.

To ensure that passwords are set correctly, reset them on both the local and remote (RSX) nodes by using the following commands:

```
NCP>SET EXECUTOR TRANSMIT PASSWORD password
NCP>SET EXECUTOR RECEIVE PASSWORD password
```

where *password* is 1 to 8 ASCII characters.

NOTE

> The transmit password on the local node must be identical to the receive password on the remote node. Some non-RSX operating systems may require a full 8-character ASCII password for connection to an RSX operating system. For VMS nodes, passwords are set on a per circuit basis.

If the circuit is still in the STARTING substate:

- Check that the hardware and software device line speed settings and duplex mode are the same on both nodes.
- On local coaxial cable links, check that the transmit cable is connected to the receive cable on the remote node.
- If a null modem cable is being used for EIA asynchronous communication, check to make sure the cable conforms to the null modem wiring diagram (see Figure A-6).
- If your device is a DL11 type, check the NETGEN specification for that device. If it was not NETGENed as a DL11-WA, that DL11 must have all EIA signals including request to send (RTS) and clear to send (CTS).

- For connecting DMP, DMV or DMR devices to a DMC device, check that all three devices are set to DMC compatibility mode. When connecting DMP, DMV, and DMR devices to other DMP, DMV, or DMR devices, both devices in the pair must have DMC compatibility mode enabled or disabled.

- Check the interface installed in your node against the types listed in Table A-1. If your interface is not listed in the table, the network software will not function correctly. For example, a DL11-F is not the same as a DL11-E and does not have full EIA functionality.

- If your device supports switch settings that supercede software settings, check these options (for example, the mode switches on DMP and DMV devices).

When the nodes are physically connected and the hardware communications devices are functioning properly, the circuit counts (bytes sent and bytes received) increment.

If the STARTING substate persists and the circuit count (bytes received) is not incremented, the hardware is probably at fault.

If DDCMP line communication has been established, the bytes sent and bytes received counters are incrementing, and line errors are not occurring, check other sources of possible software incompatibilities (for example, maximum node address or large data buffer sizes).

Local software resource allocation failures can cause line synchronization to be lost. Use the SHOW SYSTEM COUNTERS command to see if allocation failures have occurred. Then, increase the number of buffers that have allocation failures.

It is important to note that the DECnet local node software and hardware verification procedures must have been followed on both nodes (see Sections 2.1 and 2.2).

Events logged on the console help locate the problem (provided the logging console has been enabled). For more information on hardware loopback tests, refer to Appendix A.

When all lines have been verified to be in an ON state, return to Section 2.3.2 and run the test.

2.3.4 Ethernet Error Handling

The circuit state must be ON and the adjacency (the remote node on the Ethernet) must be reachable in order to run remote tests. If the adjacency is not reachable, the remote operator should make sure that the remote node's corresponding circuit has been set to the ON state.

2.3.4.1 **Problems Connecting to the Ethernet —** If the circuit is still in the STARTING substate,

- Check that the CSR and VECTOR set in the software match the CSR and VECTOR set in the hardware
- Check that the transceiver cable is connected properly on local cable links
- Perform the hardware verification described in Section A.3

When the nodes are physically connected and the hardware communications devices are functioning properly, the circuit counts (bytes sent and bytes received) increment.

Events logged on the console help locate the problem (provided the logging console has been enabled). The following event confirms when the connection to the Ethernet has occured:

```
 Event type 4.10, Circuit up
Occurred 1-Jun-83 11:22:05 on node 16 (EOWYN)
Circuit UNA-0
```

If the STARTING substate persists and the circuit count (bytes received) is not incremented, the hardware is probably at fault.

2.3.4.2 **Problems Connecting to the Adjacency —** If communication has been established, the bytes sent and bytes received counters are incrementing, and line errors are not occurring, check other sources of possible software incompatibilities (for example, maximum node address or large data buffer size).

Local software resource allocation failures can cause line synchronization to be lost. If this occurs, use the SHOW SYSTEM COUNTERS command to see if allocation failures have occured. Then, increase the numbers of buffers that have allocation failures.

It is important to note that the DECnet local node software procedures have been followed on both nodes (see Section 2.1). You should also perform the hardware verification procedures described in Appendix A to help isolate hardware problems.

If passwords have been set for your local node as well as for the adjacency that you are trying to reach, you must verify that the passwords have been set appropriately on both nodes. See Section 2.3.3 for a discussion involving problems related to passwords.

The following event confirms when the adjacency is reachable:

```
 Event type 4.15, Adjacency up
Occurred 1-Jun-83 11:22:06 on node 16 (BOSTON)
Circuit UNA-0
Adjacent node = 19 (HARTFD)
```

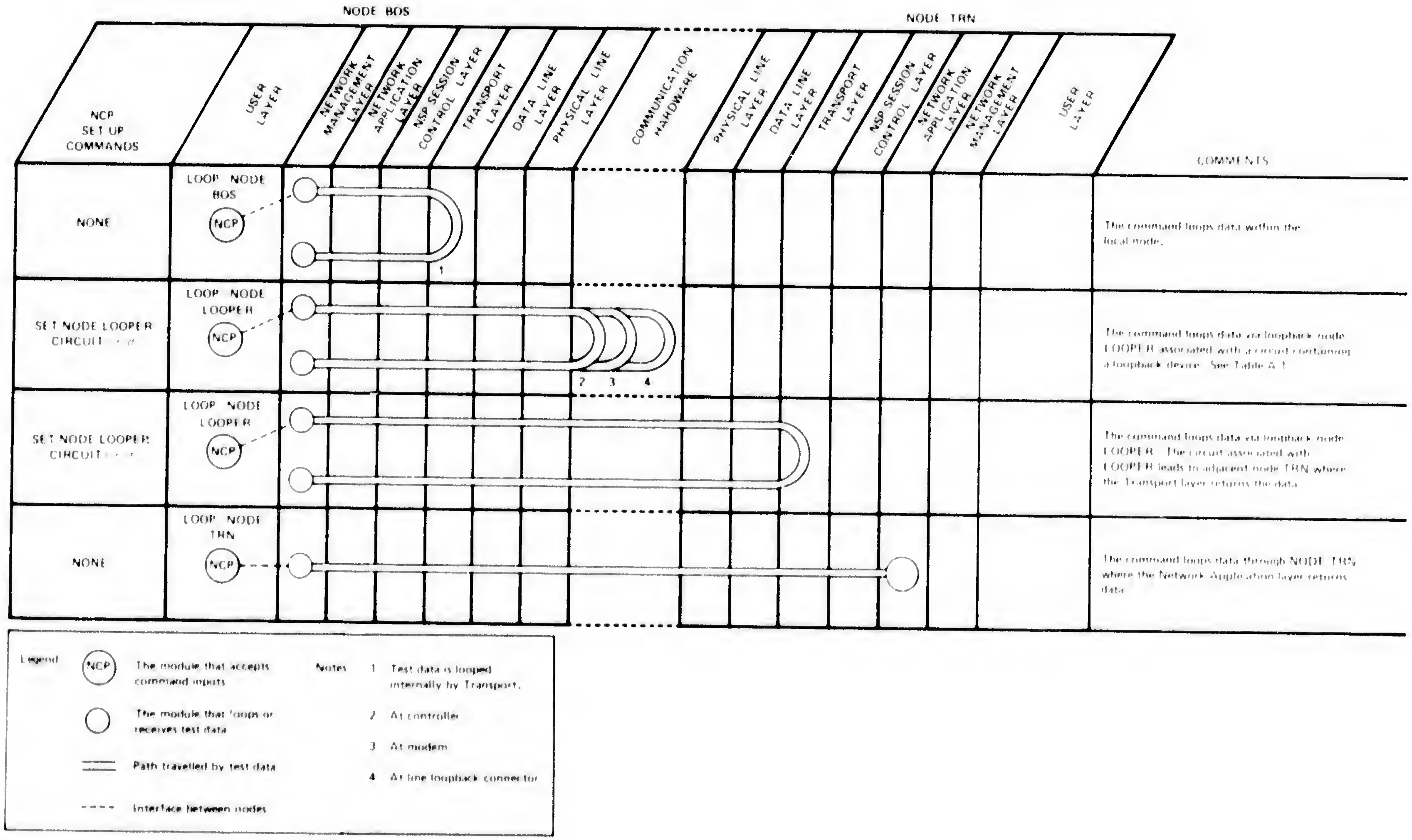


Figure 2-6: Node Level Loopback Tests Using DECnet-supplied Software

2.4 RSX-11S Testing Restrictions

Most of the tests described in this chapter can be performed on a properly configured RSX-11S node.

The NTEST.CMD procedure, however, cannot be run at RSX-11S nodes. If NTEST.CMD is run at an RSX-11M/M-PLUS node against an RSX-11S node, the file transfer tests will fail, and the DTS tests may fail depending on the configuration of the remote system. However, most of the network management commands needed to thoroughly test an RSX-11S node are available with NCP or NICE.

RSX-11S systems do not have to be configured with any network management tasks. For initial testing, it is good practice to install NCP and NICE and then later remove the tasks, if necessary.

NOTE

> For RSX-11S systems, NCP and NICE are subsets of those utilities supported in RSX-11M/M-PLUS systems.

If FAL is installed and fixed on the RSX-11S node, you can transfer data to and from unit record devices. NFT does not run on RSX-11S systems.

In addition, the DTR, TLK, and LSN tasks can be installed and fixed on RSX-11S nodes if these tasks are to be used. DTS cannot be run on RSX-11S nodes.

2.4.1 Guidelines for Using NCP and NICE for RSX-11S Systems

- **Using NCP.** NCP for RSX-11S systems does not support LOOP CIRCUIT tests.
- **Using NICE.** If you are going to use LOOP NODE with a loopback connector, the loop node can be set up with VNP or NCP.

  NICE for RSX-11S systems does not support LOOP CIRCUIT tests.

  NICE for RSX-11S systems does not allow loopback tests from remote nodes. To perform this type of test, the loopback mirror (MIR...) must be installed on the system and defined as object 25.

  NICE for RSX-11S systems does not support SHOW LINE or SHOW LINE CHARACTERISTICS, but does return information for SUMMARY, STATUS, and COUNTERS. See the *DECnet-RSX System Manager's Guide* for details on the NICE commands supported by RSX-11S nodes.

2.5 Summary

Figures 2-6 and 2-7 illustrate and summarize the kinds of tests that are available. Note that the tests can be run using network management utilities or software that runs at the User layer. For more information on running node level tests, see Appendix A and the *DECnet-RSX System Manager's Guide*.

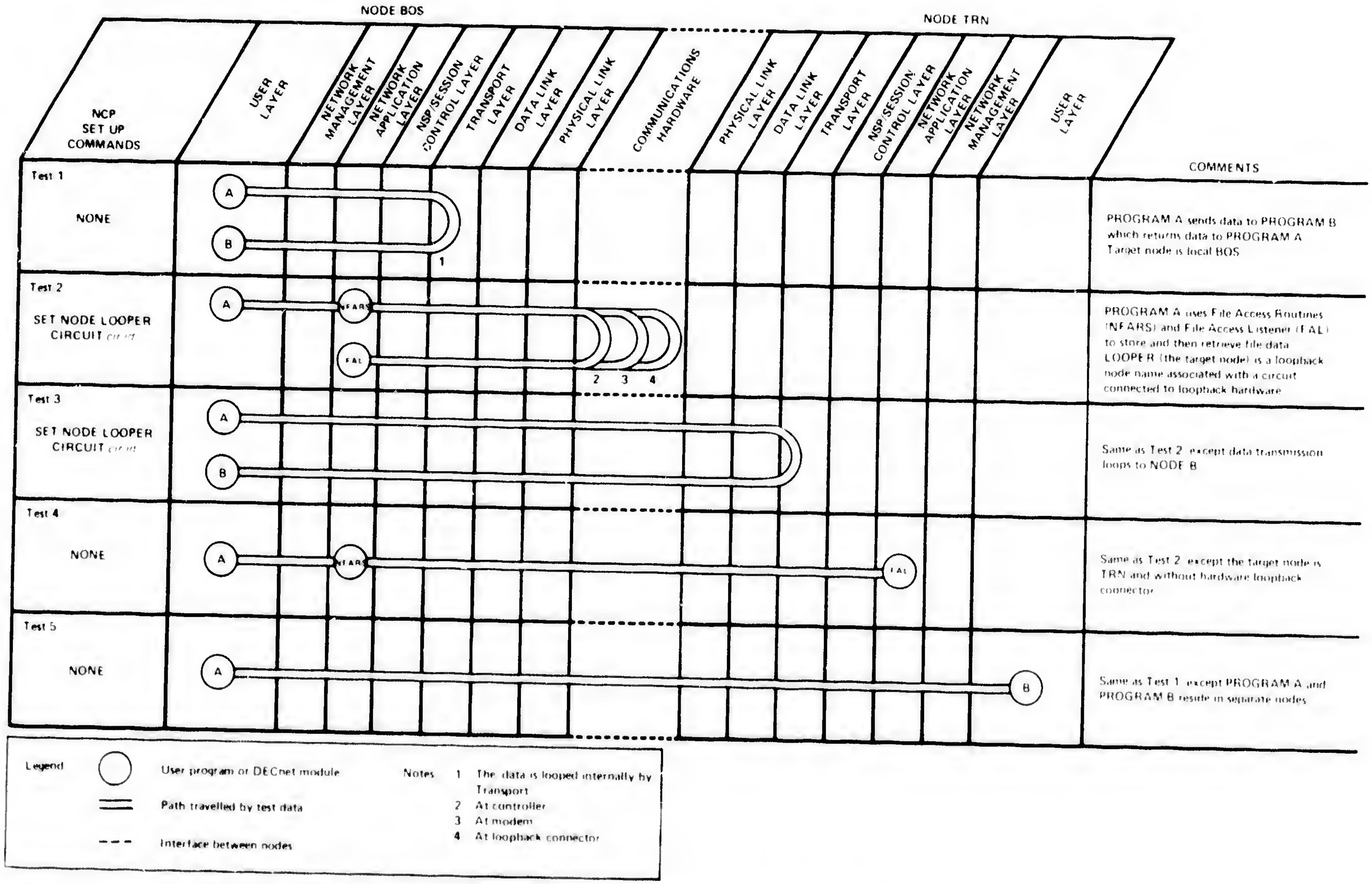


Figure 2-7: Node Level Loopback Tests Using User-supplied Software

Chapter 3
Node Configuration Guidelines

This chapter describes guidelines for configuring your DECnet node to assure proper network routing operation. The information presented here assumes that you have performed NETGEN and have verified that the node operates properly. The information in this chapter pertains to the operation of both your node and other nodes in your network. If a node in your network is other than a DECnet-RSX node, refer to the appropriate system manager's guide for information about managing that node.

3.1 Avoiding Routing Problems

To avoid network routing problems, adhere to the following guidelines when configuring nodes:

- Assign a unique address to each node.
- Specify the same maximum node address for all nodes. This will ensure that all routing tables are the same length.
- Make the large buffer size the same for all nodes in the network. This will ensure that the large buffer and segment buffer sizes are adequate for each path in the network. (NETGEN will provide the largest segment buffer size permissible for the large buffer size specified.)
- Use the same list of remote node names for all nodes.

3.2 Consequences of Ignoring the Guidelines

The likely consequences of ignoring these guidelines are:

- **Two nodes have the same address (not name).** When two nodes have the same address (not name), one of two situations can result:

  If the path costs to send messages to these nodes differ, messages are always directed to the node with the lower path cost.

  If the path costs to send messages to these nodes are the same, messages are always sent only to one of these nodes (as individually determined by system implementation).

  In both cases, messages can be received from both nodes, but can be sent only to one, as described above.

- **The maximum address on a node is larger than that on an adjacent** node. When this situation occurs, routing information is lost at the adjacent node. Those nodes whose addresses are greater than the maximum address are unreachable. The following event message occurs when a routing message is received:

```
Event type 4.5, Partial routing update loss
From node 19 (SFR) occurred 13-FEB-83 08:22:35
Circuit DUP-0
```

  A remote node is unreachable if its address is higher than that specified in the routing table.

- **The large buffer size of an intermediate node is smaller than the segment** size of the source and destination nodes. When this situation occurs, the node adjacent to that intermediate node cannot forward the packet and logs the following message:

```
Event 4.3 Oversized packet loss
From node 19 (SFR) occurred 13-FEB-83 09:39:26
Circuit DUP-0
Packet header = A 1273 1240 2
```

  If event logging is not enabled, it may appear that nothing is happening. If the mismatch is between the local node and an adjacent node, execute a SHOW NODE *node-id* COUNT command. Check the counters for oversize packet loss. An example using *node-id* SFR is given below.

```
>NCP SHOW NODE SFR COUNT

Node counters as of 13-FEB-83 12:50:03

Executor node = 19 (SFR)

      23113  Seconds since last zeroed
     653146  Bytes received
     653146  Bytes sent
      15532  Messages received
      15467  Messages sent
         65  Connects received
         65  Connects sent
          2  Response timeouts
          0  Received connect resource errors
          8  Maximum logical links active
          0  Aged packet loss
          0  Node unreachable packet loss
          0  Node out-of-range packet loss
          1  Oversized packet loss
          0  Packet format error
          0  Partial routing update loss
          0  Verification reject
```

Note that the oversized packet loss counter is 1.

To correct the problem, determine the present setting of the segment buffer size for your local and destination nodes and the large buffer size on the node that is causing the oversized packet loss. Then, either increase the buffer size of the intermediate node, or decrease the segment buffer size at the source or destination node.

On an RSX node use the NCP SHOW SYSTEM CHARACTERISTICS and SHOW EXECUTOR CHARACTERISTICS commands to determine the large buffer and segment buffer sizes:

```
NCP SHOW SYSTEM CHARACTERISTICS

System characteristics as of 13-FEB-83 12:56:16

   Maximum control buffers = 22, Maximum small buffers = 13
   Maximum large buffers = 17, Large buffer size = 280
   Minimum receive buffers = 3

NCP SHOW EXECUTOR CHARACTERISTICS

Node characteristics as of 12-JUL-83 14:31:24

Executor node = 19 (ELROND)

Identification = DISTRIBUTED SYSTEMS, Management version = 4.0.0
Host = 19 (ELROND), Loop count = 1, Loop length = 40
Loop with = mixed, NSP version = 4.0.0
Maximum links = 20, Routing version = 2.0.0, Type = Routing IV
Routing timer = 300
Maximum address = 300, Maximum circuits = 8, Maximum cost = 1022
Maximum hops = 12, Maximum visits = 24
Maximum broadcast nonrouters = 32
Maximum broadcast routers = 10
Segment buffer size = 576
Verification state = On
```

On a VAX node use the NCP SHOW EXECUTOR CHARACTERISTICS command to determine the large buffer size.

Now, increase the large buffer size on the intermediate node with the following command:

```
CFE DEFINE SYSTEM LARGE BUFFER SIZE 512
```

With the DEFINE command above, you have increased the buffer size from 280 bytes to 512 bytes. The acceptable range for buffer size is from 192 to 1484 bytes.

An alternative to increasing the large buffer size is to reduce the executor segment size with the following command:

```
CFE DEFINE EXECUTOR SEGMENT BUFFER SIZE 262
```

This command decreases the segment buffer size from 576 to 262. (The acceptable range for segment buffer size for DECnet-RSX is from 174 to 1466 bytes.) Remember that the large data buffer size must be at least 18 bytes greater than the segment buffer size. NETGEN initially sets the segment buffer size to be 18 bytes less than the large data buffer size, and CFE will also check for this 18 byte difference.

NOTE

> Changes to the large or the segment buffer size are made to the permanent data base. These changes are not reflected in the running data base until a CLEAR SYSTEM and a SET SYSTEM command are performed.

Chapter 4
Testing the RSX-11 PSI Installation

This chapter describes the procedure to be observed by users of a combined DECnet/PSI configuration. The procedure verifies that the RSX-11 PSI system has been generated and installed correctly. The chapter also describes how you can test the X.29 software to ensure that it has been installed correctly. Section 4.3 contains information on verifying data link mapping (DLM) circuits.

4.1 The Installation Checkout Procedure

The procedure uses two programs: XTR (X.25 test receiver) and SCPXTS (X.25 test sender). They verify that your PSI system can set up a call and transfer data between your local DTE and a remote DTE. SCPXTS is provided as a prebuilt task in your distribution kit, and it communicates with XTR running on a remote system.

4.1.1 Preparing to Run SCPXTS

Before you can run SCPXTS, you must have generated and installed your RSX-11 PSI system. A checklist of the steps you must have performed is given below, and full details can be found in the *RSX-11 PSI Generation Guide*:

- Run SYSGEN (if necessary)
- Run PREGEN
- Run NETGEN
- Install the system

Once you have performed these steps, the RSX-11 PSI system is up and running. To run SCPXTS, your generated system must subscribe to the open user group and must have at least one switched virtual circuit (SVC).

4.1.2 Running SCPXTS

To run SCPXTS, use the following format:

>RUN *ddu:[uic]*SCPXTS

where *ddu:[uic]* is the location of SCPXTS after NETGEN (see the *RSX-11 PSI Generation Guide*). SCPXTS asks for the address and subaddress of the remote DTE, as follows:

```
Input full DTE address :
```

The answer must consist of the 1- to 14-digit DTE address immediately followed by the 0- to 4-digit subaddress.

The remote DTE must be a machine running XTR under the PSI product. This can be one of your own machines or any other suitable machine. If you are not using one of your own machines, enter the address provided by your field software support specialist.

SCPXTS will then run for 5 to 15 minutes, depending upon the configuration of your system and on any other tasks running at the same time. The procedure tests the installation of the RSX-11 PSI product by setting up a call between SCPXTS and XTR. These programs transmit and receive 500-byte long data packets continually for three minutes. When the tests stop, t e call is cleared. When the tests have completed, SCPXTS produces a summa y. An example of a summary from a normal run follows:

```
Test Finished

                         Test Statistics
          Number of transmits     =          180
          Number of receives      =          180
          Interrupts completed    =          0
          Interrupts received     =          0
          Interrupts confirmed    =          0
          Resets completed        =          0
          Resets received         =          0
```

SCPXTS and XTR can run at different speeds. Therefore, the number of transmits and receives reported by the SCPXTS summary may not be equal. However, the number of transmits and receives should always be approximately equal and should exceed 170. A summary showing fewer than 170 transmits and receives does not mean that the test procedure failed, but it could imply PSN problems such as noisy lines or network congestion.

4.1.3 Error Messages

If your PSI system is not working correctly, SCPXTS prints error messages in the following format:

%ICP-E-Error*x* occurred

where *x* is an error number. The following error numbers can occur:

Number	Meaning
1	Error in address/user group Information message. The address of the remote DTE was not specified correctly.
2	Could not send set-up message Fatal error. A data transmission error occurred, or SCPXTS failed to reach XTR.
3	Set-up message was not same when polled Fatal error. A data transmission error occurred, or SCPXTS failed to reach XTR.
7	Could not assign network device This error can occur if XTR is not set up correctly. Try the test again in case the network was the cause of the error, and if the error recurs, ask your software support specialist to make sure that XTR is fully operational.
20	Received diagnostics code not as expected Warning message. The consistency check on a clear or reset diagnostics code failed. This means the clear or reset was generated by the network.
21	Received cause code not as expected Warning message. The consistency check on a clear or reset cause code failed. This means the clear or reset was generated by the network.
22	Data received not as expected Warning message. The consistency check on incoming data failed due to a data transmission error.
23	Clear data not as expected Warning message. The consistency check on the clear data failed. This means the clear was generated by the network.
24	Failed to set up with XTR Fatal error. SCPXTS failed to reach XTR.

You are advised to take advantage of the extensive status and error counters maintained by RSX-11 PSI to aid in diagnosing problems (see the *RSX-11 PSI System Manager's Guide*). You may also need to test the physical line using loopback tests (see the *RSX-11 PSI System Manager's Guide*).

For more details on SCPXTS, see the *RSX-11 PSI Generation Guide*.

4.2 Verifying X.29 Installation

To verify that the X.29 terminal driver has been correctly installed, use a portable terminal and acoustic coupler. Perform the following steps:

1. Dial an X.29 PAD (packet assembly/disassembly) facility.
2. Log into an account on the local DTE.
3. Engage the DTE in a simple dialog by using a few standard commands, such as:

   PIP/LI

   EDI

4. Log out.

See the *RSX-11 PSI Network-specific Information* manual for further details on these steps.

4.3 Verifying Data Link Mapping Installation

Once you have established that your local DTE is fully operational by following the procedures described in Section 4.1, check any DLM circuits that may be in the network in the following manner:

1. Ensure that the DTE is ON. Enter the following NCP command:

   `NCP SHOW MODULE X25-PROTOCOL DTE` *dte-address*

   The system response will show you whether the DTE you are testing for is ON or OFF. If it is OFF, turn it ON by means of the NCP command:

   `NCP SET MODULE X25-PROTOCOL DTE` *dte-address* `STATE ON`

2. For PVCs, ensure that the channel numbers, maximum window size, and maximum packet size defined are the same as that allocated to you by the PPSN. To determine this information, enter the following CFE command:

   `CFE>LIST CIRCUIT DLM-`*x.y* `CHARACTERISTICS`

3. For SVCs, check that the usage parameters for the DLM circuits are consistent on each node (one side must be outgoing and the other side incoming). Enter the following CFE command for each circuit:

   `CFE LIST CIRCUIT DLM-`*x.y* `CHARACTERISTICS`

   The node response will show whether the circuit is incoming or outgoing. If a correction is required, effect it by means of the following CFE command for the appropriate circuit:

   `CFE DEFINE CIRCUIT` *circuit-id* `USAGE` *usage*

   where usage = INCOMING or OUTGOING

For SVCs you should also verify whether or not the maximum packet size and maximum window size will be negotiated by the software. If the packet size and/or window size is a value other than zero, then either one (or both) of these will be negotiated. The default is zero (which implies no negotiation with the network). Therefore, if your network does not support negotiations, the values for the packet size and/or window size must be zero. Use the following commands to specify the maximum packet and window sizes:

```
CFE DEFINE CIRCUIT DLM-x.y MAXIMUM DATA byte-count
CFE DEFINE CIRCUIT DLM-x.y MAXIMUM WINDOW block-count
```

4. The subaddress of the outgoing circuit must be within the range specified in the PSI portion of NETGEN in response to the question

   ```
   WHAT IS THE TRANSPORT SUBADDRESS RANGE?
   ```

   If it is not, correct it by entering the CFE command

   ```
   CFE DEFINE CIRCUIT DLM-x.y NUMBER dte-address
   ```

   The variable *dte-address* consists of up to 15 digits made up of two parts:

   - The DTE's unique address within the network.
   - Subaddress.

   The length of *dte-address* depends on the PSN to which you are connecting. The subaddress is the low-order portion of the *dte-address*.

5. In testing SVC circuits, be aware of the possibility of exceeding the maximum number of re-calls. If this number is exceeded, the circuit will go to the ON-FAIL state. This would be reported in response to the NCP command:

   ```
   NCP SHOW CIRCUIT circuit-id
   ```

   If the circuit is ON-FAIL state, first turn it OFF and then ON by using the following NCP commands:

   ```
   NCP SET CIRCUIT circuit-id STATE OFF

   NCP SET CIRCUIT circuit-id STATE ON
   ```

   Once the circuit has been set to the ON state, run NTEST.CMD as a remote node test (see Section 2.3).

Appendix A
Line/Circuit Testing

Line/circuit tests are a set of tests that can be used to isolate line, circuit, and device problems.

Line/circuit level loopback tests confirm a physical link by looping test data in the following ways:

- Through a hardware loopback device on the line, through a modem (or loopback connector), or through a remote node. The tests conducted using a loopback connector or a modem are called line/circuit loopback tests.
- By using the software of a remote system. These are called software loopback tests.
- By performing Ethernet loopback tests that loop messages to remote devices on the Ethernet.

You may want to initiate a series of operations to test various aspects of a physical link-for example,

- To attach a loopback connector to the controller and test to see if the line is functional using a line loopback test (see Section A.1)
- To use a modem with analog loopback capability to see if the line is functional (see Sections A.1.1 and A.1.1.1)
- To perform a loopback test to another node to see if the physical line is operational (called a software loopback test) (see Sections A.2 and A.3)
- To perform Ethernet loopback testing to see if remote devices on the Ethernet are able to transmit and receive messages (see Section A.3)

Use the NCP LOOP CIRCUIT command to perform a line/circuit level loopback test. When you issue this command, you have the option of controlling the type of binary information (mixed, ones, zeroes), the number of blocks of information (1 to 65535), and the length in bytes of each block (1 to 65535) to be looped. Line level hardware-looped tests cannot be performed with DMC and DMR devices because of restrictions unless a software loopback mirror provides the loopback on a remote system (that is, you cannot use a hardware loopback connection). See the *DECnet-RSX System Manager's Guide* for the complete syntax of the LOOP CIRCUIT command.

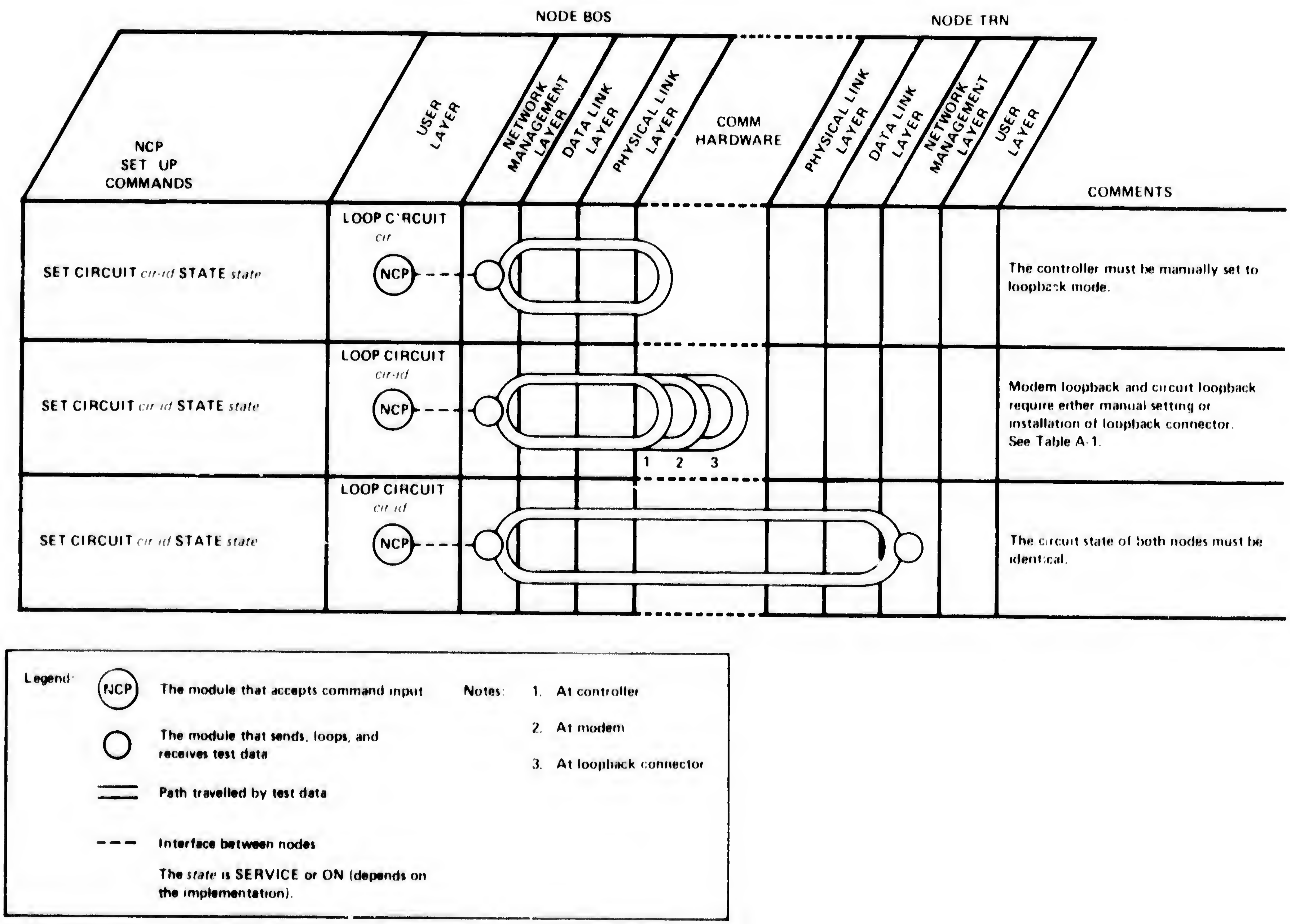


Figure A–1: Line Level Tests

If your message returns with an error, the test stops, and NCP prints a message indicating a test failure, the reason for the failure, and a count of the messages that were not returned (see the *DECnet-RSX System Manager's Guide* for the NCP error message summary) — for example,

```
NCP>SET CIRCUIT DUP-0 STATE SERVICE
NCP>LOOP CIRCUIT DUP-0 COUNT 10

NCP -- Loop failed, line protocol error
       Unlooped count = 8
```

In this test, 10 messages were sent. The first two messages were sent successfully, and an error occurred on the third. When the error occurred, the test halted. Figure A-1 illustrates both the hardware and the software loopback tests. Refer to the *DECnet-RSX System Manager's Guide* for more information on running these tests.

A.1 Line/Circuit Level Test Using a Loopback Connector

This type of test verifies whether or not the device and cable are functional. Figure A-2 illustrates a line loopback test with a loopback connector. See Figure A-3 for other hardware loopback arrangements.

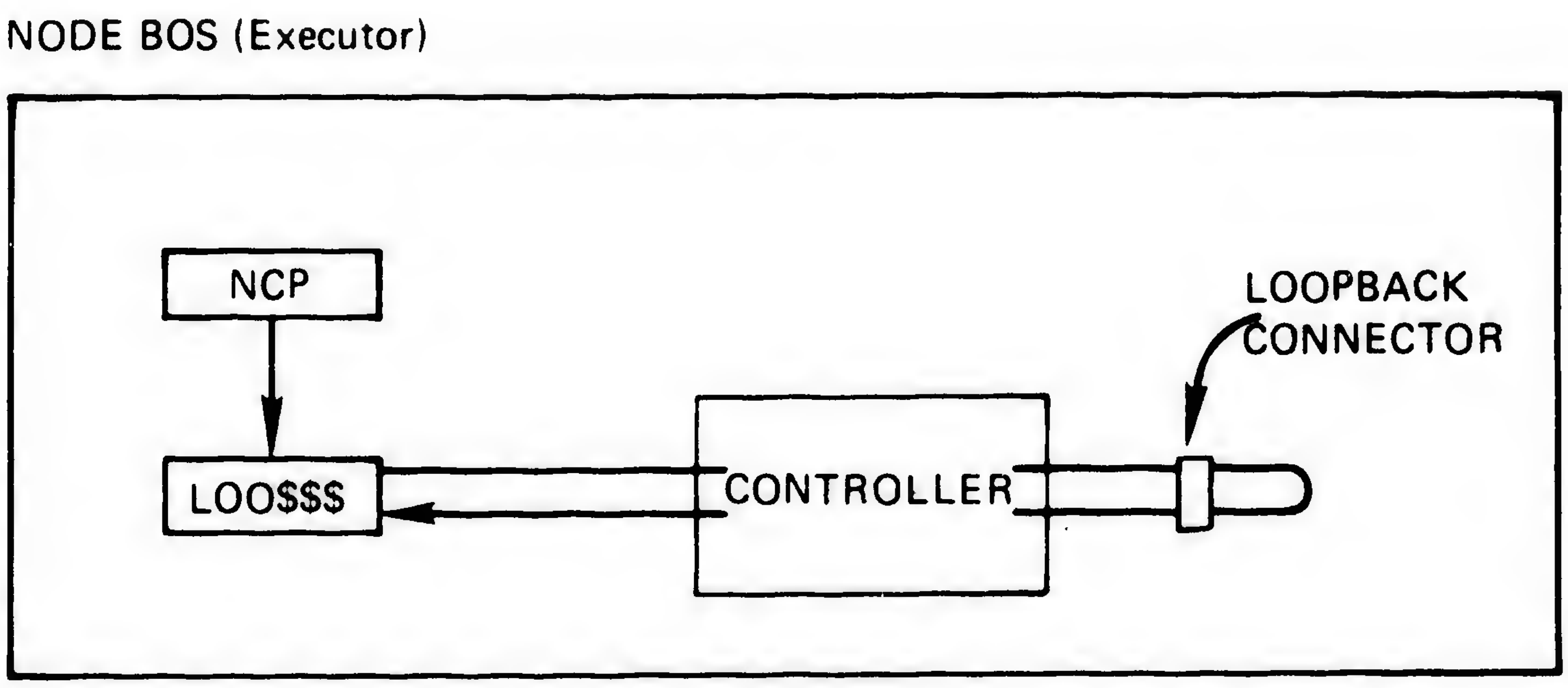


Figure A-2: Full Duplex Line Loopback Testing

For this test, put the circuit in the SERVICE or the ON state. (The test can be started from either the ON or the SERVICE state. However, the SERVICE state is the preferred state for this test, as it allows testing that does not affect the network as a whole.) Then use the LOOP CIRCUIT command — for example,

```
NCP>SET LINE DUP-0 DUPLEX FULL
NCP>SET CIRCUIT DUP-0 STATE SERVICE
NCP>LOOP CIRCUIT DUP-0 COUNT 10
NCP>SHOW CIRCUIT DUP-0 COUNT
```

This set of commands tests the physical line DUP-0 up to the loopback connector. A user can exercise the local node's hardware, including the communications controller and the cable, by using a loopback device on the line during a line level loopback test. Certain synchronous communications interfaces do not supply a free running clock for the EIA (Electronic Industries Association) interface (Table A-1). This clock must be present to run a line loopback procedure. Therefore, if the device being used does not provide a clock, a clocked null modem (modem eliminator) must be installed between the modem cable and the loopback connector. The modem eliminator supplies a clock to resolve this problem. See Figures A-4, A-5, and A-6 for wiring diagrams of the loopback connectors and the null modem connection.

Table A-1: Digital Communications Interface Specifications for Loopback Processing

Interface Name	Interface Type	External Loopback Connector	External Loopback Capability	Cables Required†
DU11-DA	EIA	H315	Needs a clocked null modem	BC99S
DUP11-DA	EIA	H325	Needs a clocked null moden.	BC02C
DV11	EIA	H325	Needs a clocked null modem	BC05D
DMC11-AL*	Local sync	12-12528 coaxial test connector	Yes**	BC03N
DMC11-AR with DMC11-DA line unit	RS232-C/ CCITT V.24	H325	Needs a clocked null modem**	BC05D
DMC11-AR*** with DMC11-FA line units	CCITT V.35/DDS	H3250	Needs a clocked null modem**	BC05Z
DMP11-AA	RS-232-C/ RS-423-A	H325 H3251	Yes	BC55C-10, BC05D-25
DMP11-AA	RS-232-C/ RS-423-A	H325 H3251	Needs a clocked null modem	BC55C-10 and null modem

(continued on next page)

* With DMC11-MA or DMC11-MD line units.

** A LOOP CIRCUIT command does not work with this interface. Use LOOP NODE.

*** For RSX systems only.

† The cables used depend on the type of modem in the system.

Table A-1 (Cont.): Digital Communications Interface Specifications for Loopback Processing

Interface Name	Interface Type	External Loopback Connector	External Loopback Capability	Cables Required†
DMP11-AB	CCITT V.35	H3250	Yes	BC05Z-25
DMP11-AC	Local sync at 56K bps or 1Meg bps	Half duplex switch ON and cables removed	Yes**	BC55A-10 and Twinax cables
DMP11-AE	RS-422-A	H3251	Yes	BC55B-10, BC55D-33
DMR11-AA with Bell 200 modem or equiv	EIA RS232-C/ CCITT V.24	H325	Yes**	BC05D 25 pin
DMR11-AA	EIA RS449/423	H3251	Yes**	BC55D 37 pin
DMR11-AB*** with Bell 500AL1/5 or equiv	CCITT V.35/DDS	H3250	Yes**	BC05Z-25
DMR11-AC	Local sync 4 selectable speeds up to 1M bps	Half duplex switch ON and cables removed	Yes**	BC03N, BC55M, or BC55N
DMR11-AE with Bell 303C type modems	EIA RS449/322	H3250	Yes	BC05Z-25
DL11-E	EIA	H315	Yes	Cable supplied in kit
DLV11-E (LSI-11)	EIA	H315	Yes	BC05C
DUV-DA (LSI-11)	EIA	H315	Needs a clocked null modem	BC05C
DZ11-A,B,H	EIA	H325	Yes	BC05D
DZV11 (LSI-11)	EIA	H325	Yes	BC11U
PCL11	Parallel	Not required	Yes	BC20K, BC20P

* With DMC11-MA or DMC11-MD line units.

** A LOOP CIRCUIT command does not work with this interface. Use LOOP NODE.

*** For RSX systems only.

† The cables used depend on the type of modem in the system.

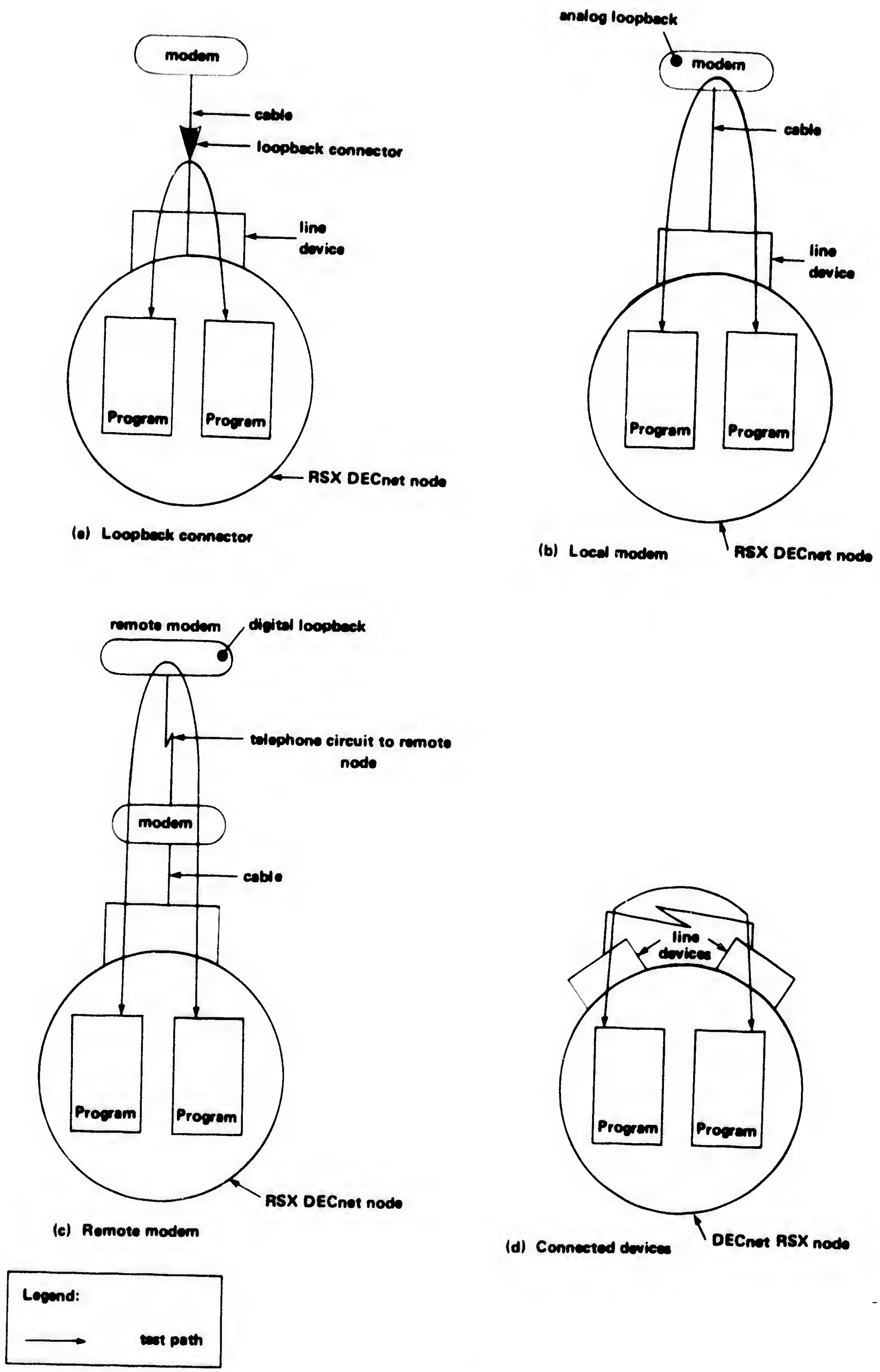
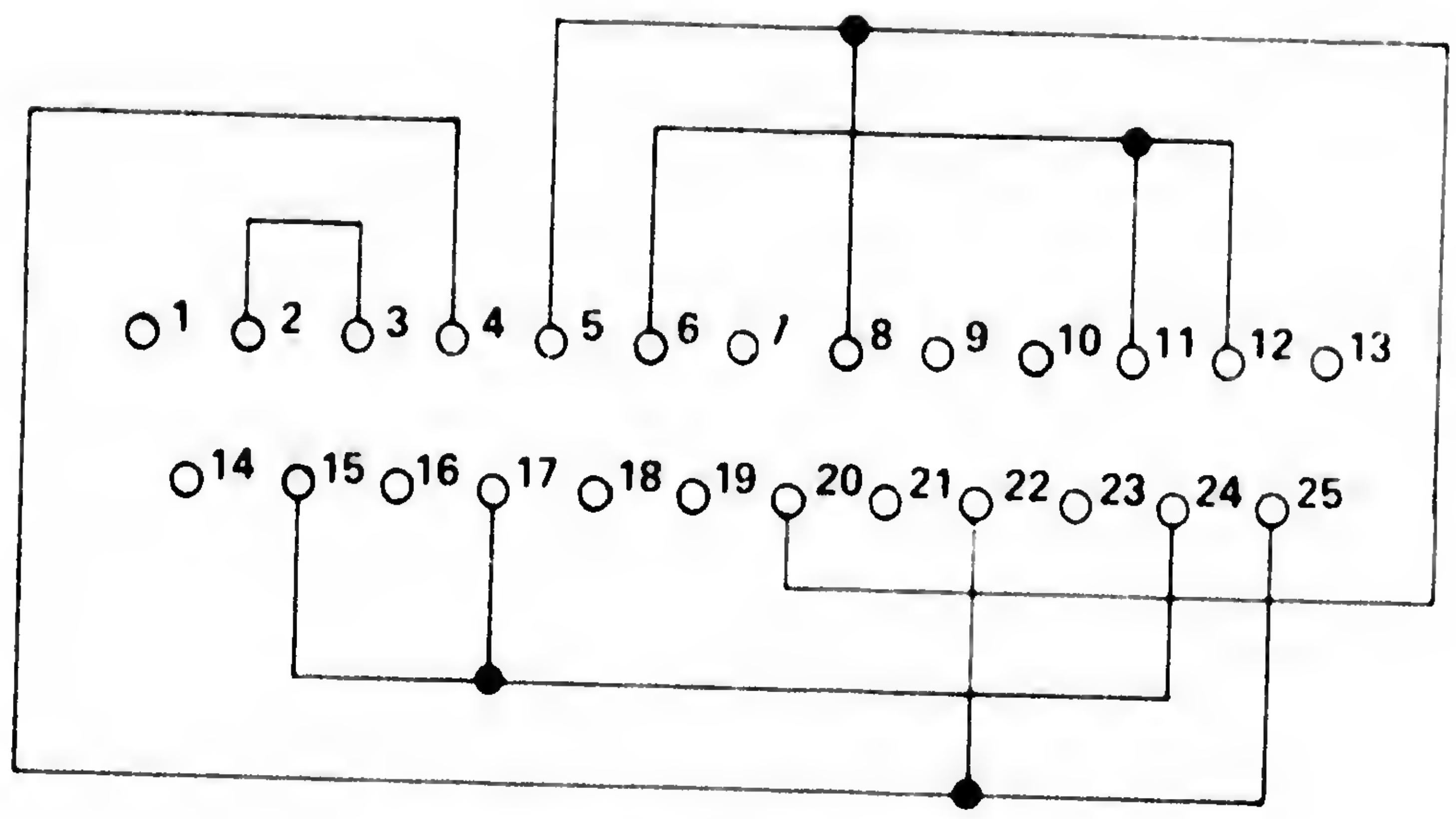


Figure A-3: Hardware Loopback Arrangements

A.1.1 Hardware Arrangements for Line/Circuit Loopback

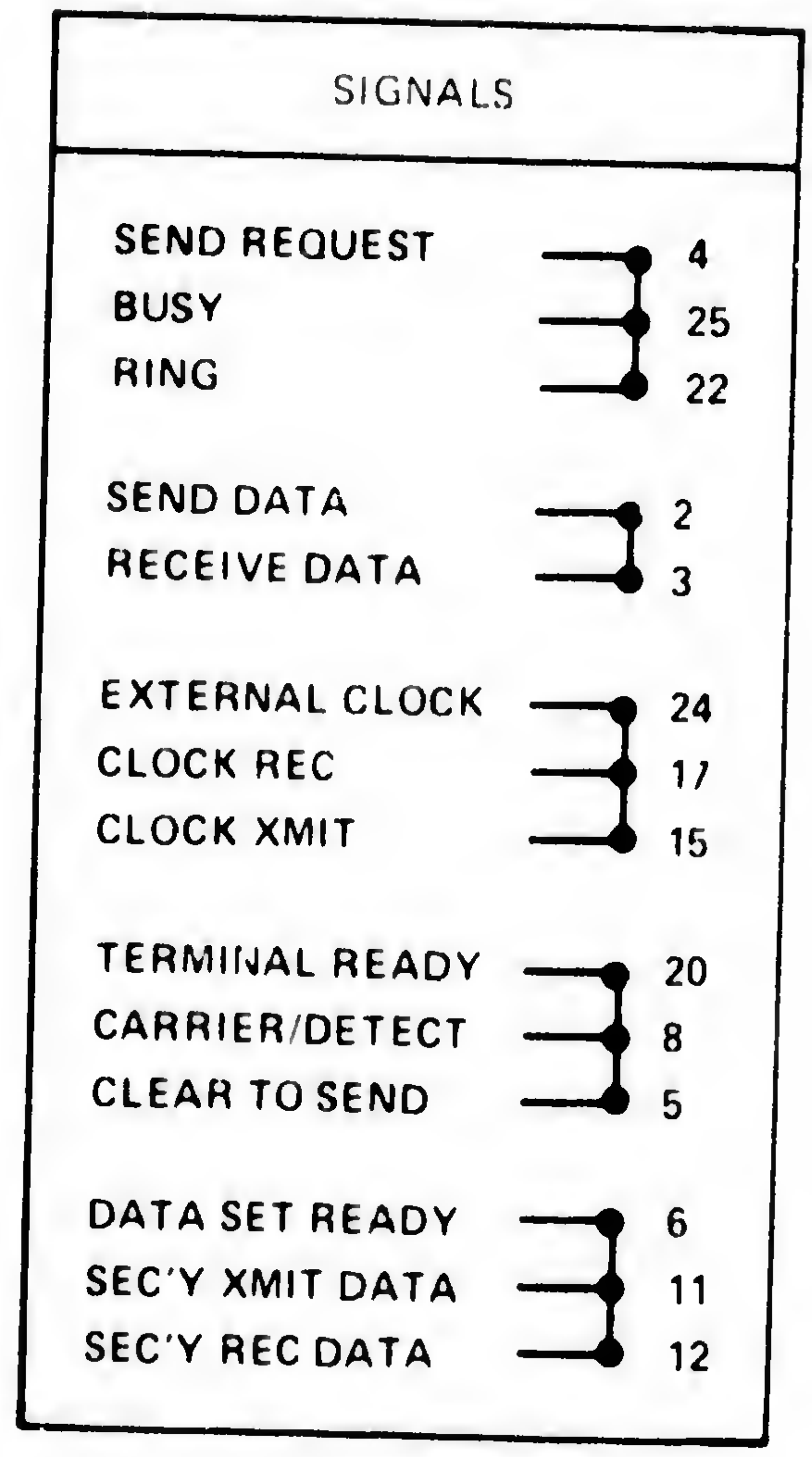
The hardware arrangement for line loopback depends on your installation's interface equipment and the hardware components you wish to include in the test.

- **Loopback connector.** The loopback connector is installed on the cable between the line device and the modem, as shown in Figure A-3(a). This configuration tests the line device and the cable. The connector returns digital data to the line device. You must condition the line device for full duplex mode by using the SET LINE DUPLEX FULL command.
- **Connected devices.** If you have more than one similar line on your system, you can use two lines for line loopback by connecting their line devices, either directly by way of coaxial cable or over a telephone circuit by way of modems as shown in Figure A-3(d). This loopback arrangement tests both line devices and whatever cables and modems connect them. The data is not forced back in the hardware, but received by a device at the local node instead of at a remote node. This is analogous to having two telephones in your house and using one to call the other. This is the only loopback arrangement whereby you can run the line devices in half duplex mode. (Full duplex mode is also permitted.)
- **Local modem loopback.** Some modems, for example 208A and 208B types, have an analog loopback (AL) button. The button directs the modem's transmitter output to its receiver input. This loopback arrangement tests the line device, the cable, and the modem, as shown in Figure A-3(b). The modem converts data from the device to analog form, digitizes the data again, and returns it to the device. You must condition the device for full duplex mode by using the SET LINE DUPLEX FULL command. Note that local modem loopback must always operate in full duplex mode, even for 208B type modems (whose telephone circuits are half duplex).
- **Remote modem loopback.** Some modems, for example 208A types, have a digital loopback (DL) button. The button directs the modem's receiver output to its transmitter input. A remote modem set in the DL state and connected to the local modem by a full duplex telephone circuit is shown in Figure A-3(c). This loopback arrangement tests the device, the cable to the local modem, the local modem, the telephone circuit, and the remote modem. The remote modem digitizes data from the local modem, converts the data to analog form again, and returns it to the local modem (and then to the device). You must condition the device for full duplex mode by using the SET LINE DUPLEX FULL command.



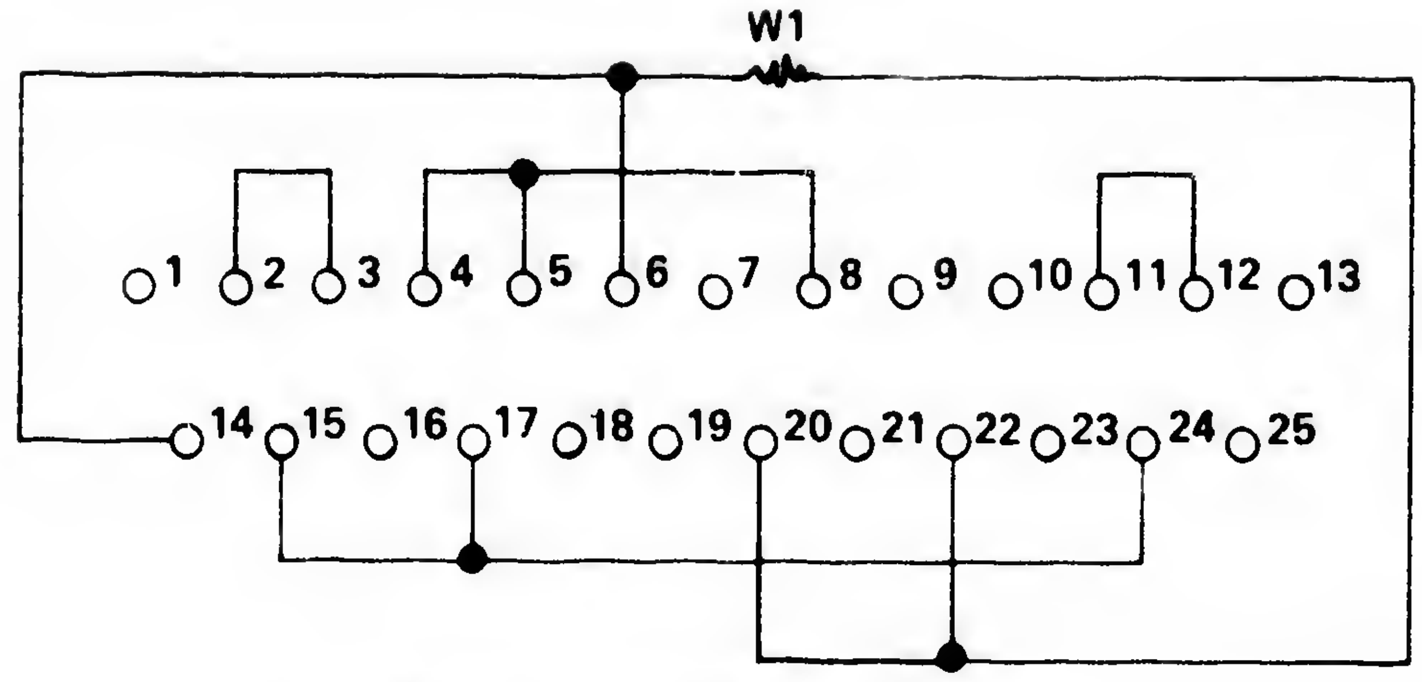
Use this connector to test DQ, DU, and DL11-E interfaces.

View Looking at Back of Connector



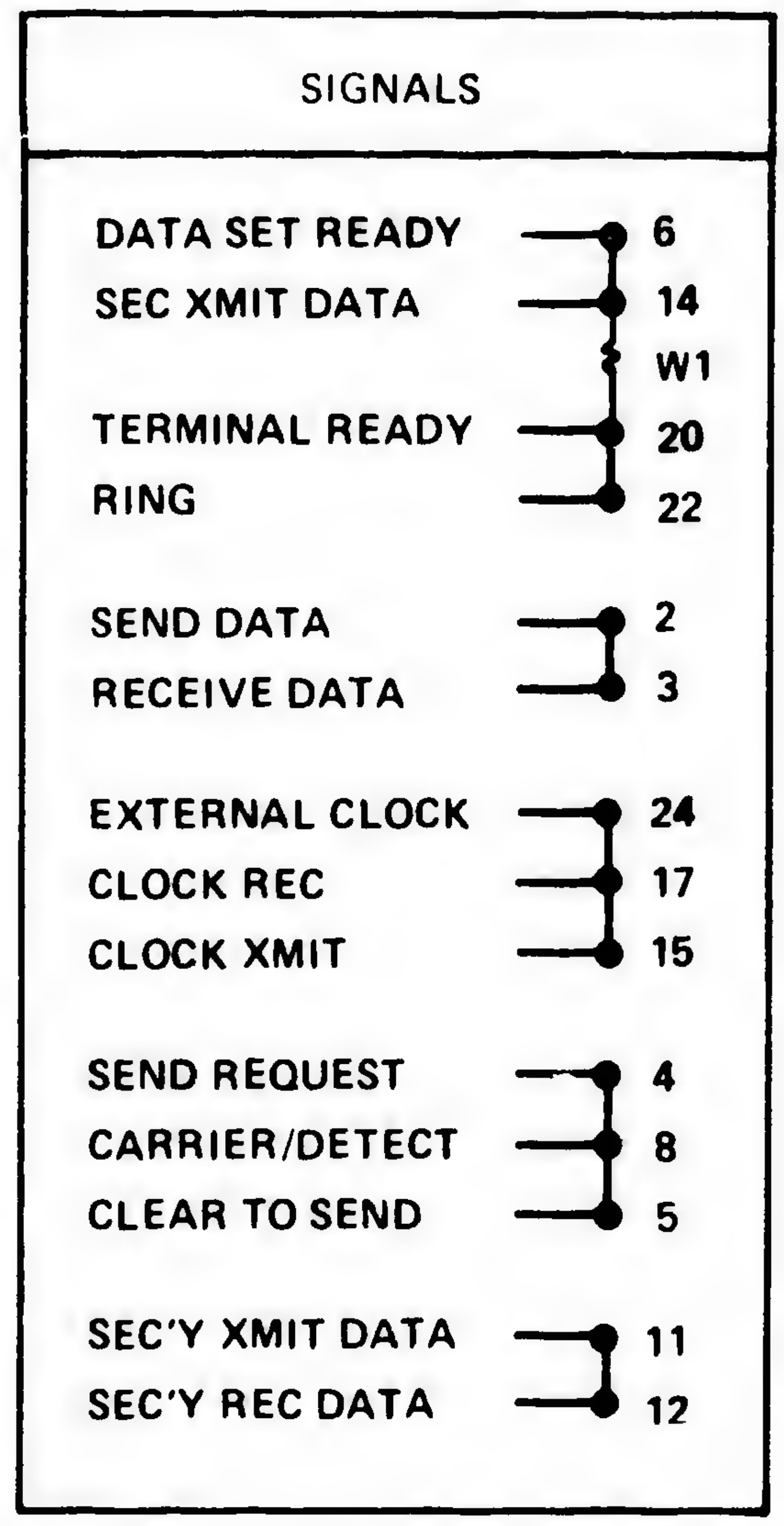
TERMINAL	SIGNAL
2	SEND DATA
3	RECEIVE DATA
4	SEND REQUEST
5	CLEAR TO SEND
6	DATA SET READY
8	CARRIER/DETECT
11	SEC'Y XMIT DATA
12	SEC'Y REC DATA
15	CLOCK XMIT
17	CLOCK REC
20	TERMINAL READY
22	RING
24	EXTERNAL CLOCK
25	BUSY

Figure A-4: H315 Loopback Connector Wiring Diagram



Use this connector to test DV, DUP, DZ, and DMC interfaces. When used with the DV11 interface, remove W1.

View Looking at Back of Connector



TERMINAL	SIGNAL
2	SEND DATA
3	RECEIVE DATA
4	SEND REQUEST
5	CLEAR TO SEND
6	DATA SET READY
8	CARRIER/DETECT
11	SEC'Y XMIT DATA
12	SEC'Y REC DATA
14	SEC XMIT DATA
15	CLOCK XMIT
17	CLOCK REC
20	TERMINAL READY
22	RING
24	EXTERNAL CLOCK

Figure A-5: H325 Loopback Connector Wiring Diagram

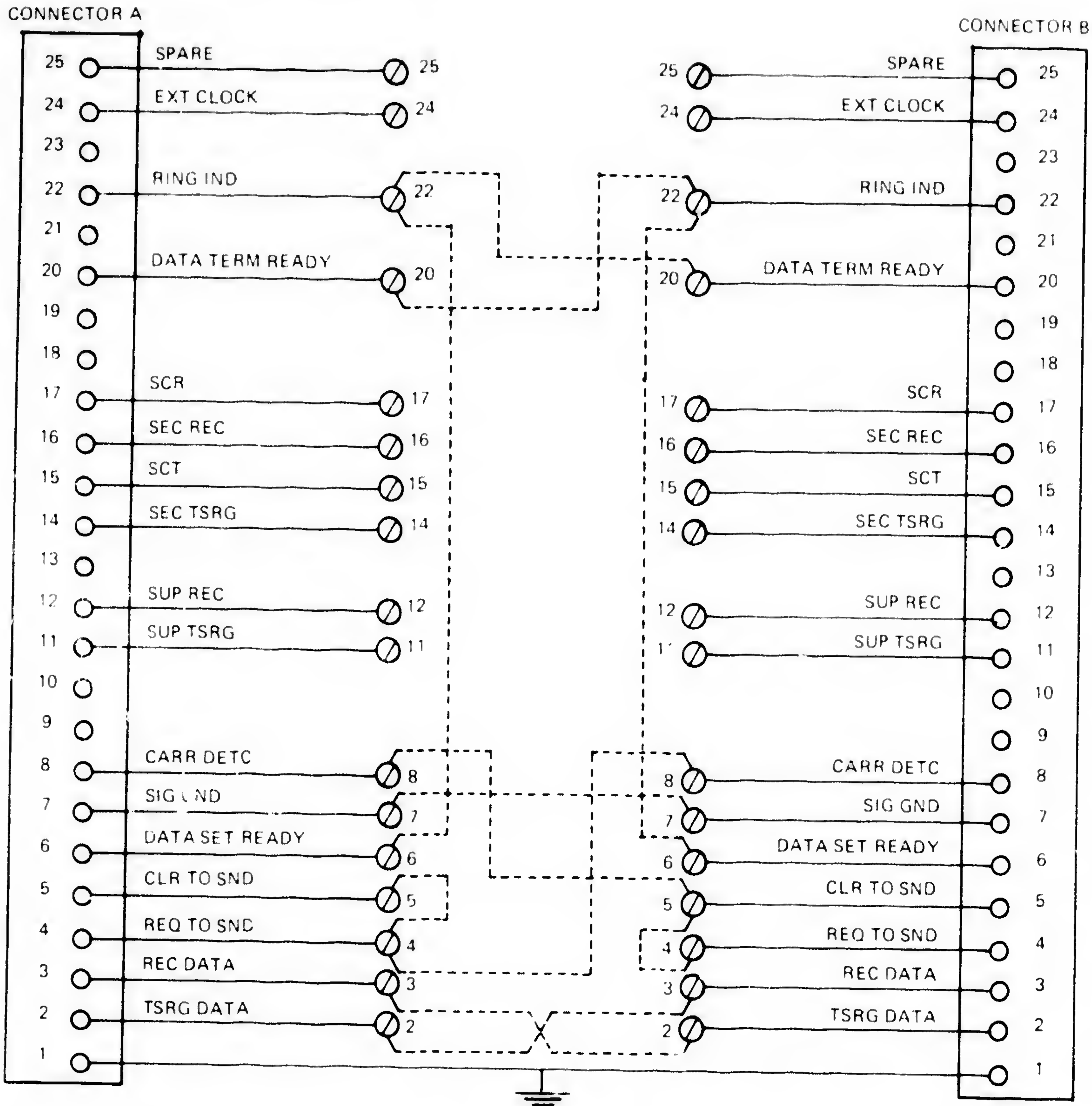


Figure A-6: EIA Asynchronous Null Modem Wiring Diagram

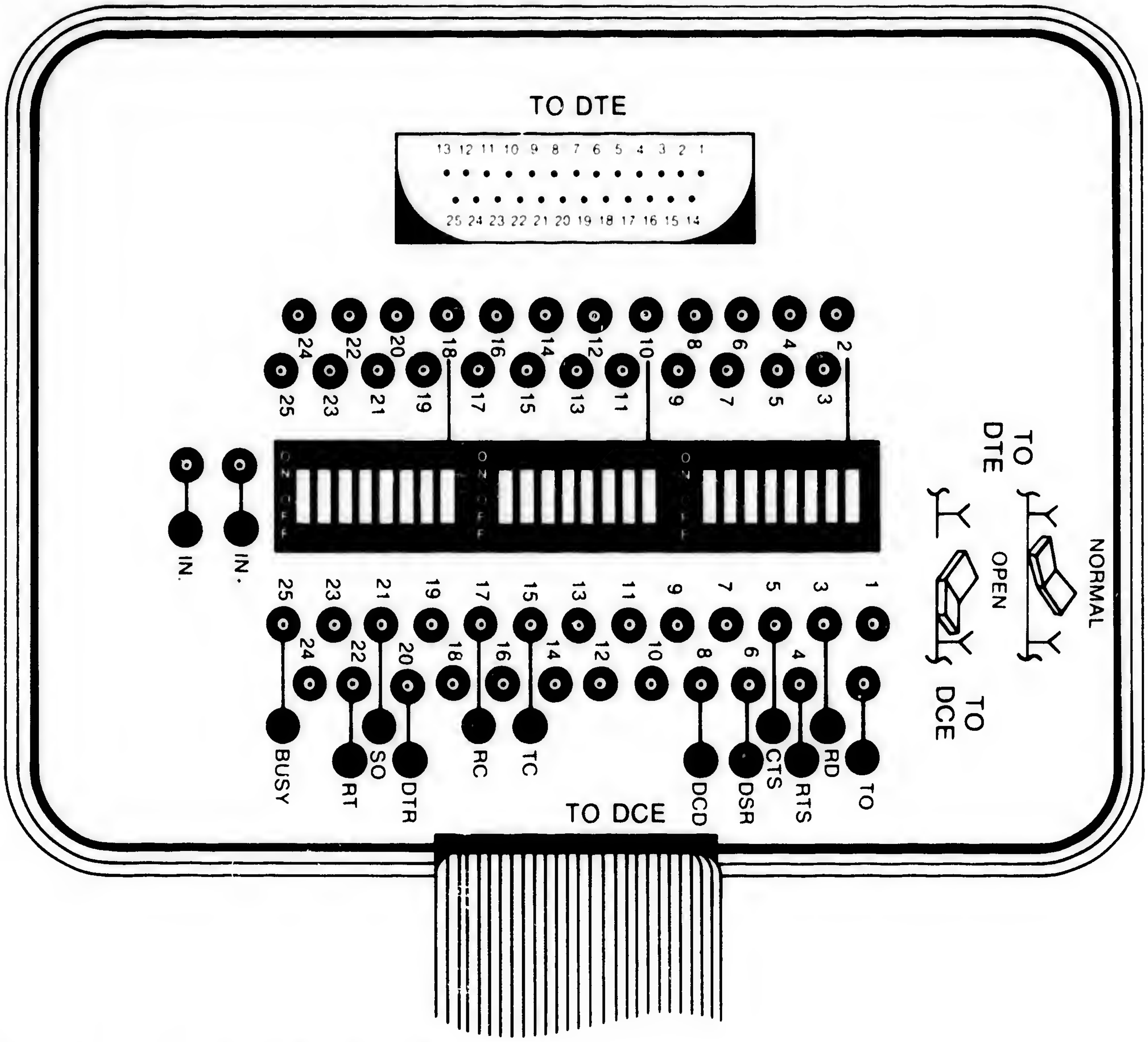


Figure A-7: Typical Breakout Box

A.1.1.1 Line/Circuit Level Test Using a Modem — In order to use a modem in a line level loopback test, the modem must have the analog loopback options. A breakout box (Figure A-7) may be required to allow the loopback of specific modem handshaking circuits that are normally set OFF when the modem is in an analog loopback state. (The communications software requires that DATASET-READY be ON. However, some modems, the Bell 201C for example, set DATASET-READY to OFF when in the analog loopback state.)

When installing the breakout box between the modem cable and the modem, make the following patches across the DTE side of the breakout box:

- Patch DATASET-READY (pin 6) to DATATERMINAL-READY (pin 20).
- Patch REQUEST-TO-SEND (pin 4) to CLEAR-TO-SEND (pin 5).

Set switches 5 and 6 on the breakout box to OFF. Set the remaining switches to ON. The DTE side of the breakout box is the side connected to the computer (as opposed to the side connected to the modem, which is called the DCE side).

A.2 Software Loopback Test (non-Ethernet devices)

Use the LOOP CIRCUIT command to perform a software loopback test of a physical line connected to the local node. This type of test uses DECnet-RSX software to loop through the physical line to circuit service software in the adjacent node and back to the local node. Figure A-8 illustrates a software loopback test to check whether or not the circuit is operational up to the remote unit and controller on the adjacent node.

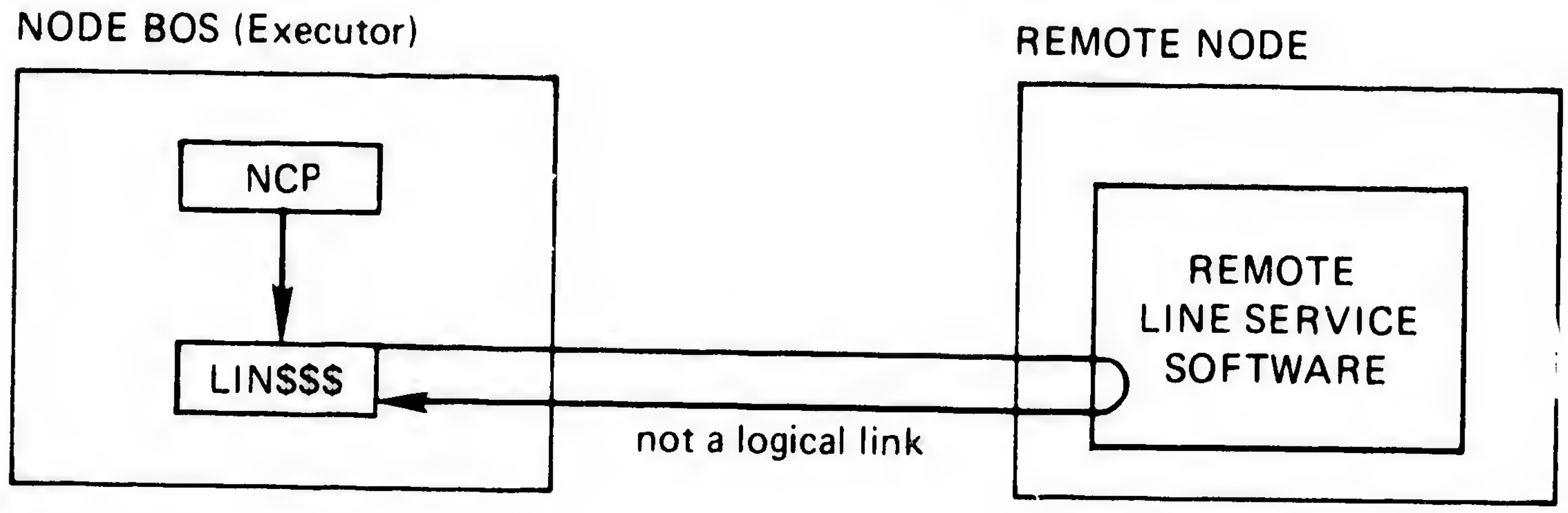


Figure A-8: Software Loopback Testing

For this test, you first turn the circuit ON, then use the LOOP CIRCUIT command:

```
NCP>SET CIRCUIT DUP-0 STATE ON
NCP>LOOP CIRCUIT DUP-0 COUNT 10
```

This set of commands tests the physical line DUP-0 up to the adjacent node.

A.3 Ethernet Loopback Testing

Ethernet loopback testing is performed in order to test remote devices on the Ethernet. Ethernet circuit loopback testing can be performed concurrently with other DECnet operations on each node and does not interfere with other Ethernet traffic. There are two ways of testing other UNAs on the Ethernet:

- Loop to any node on the Ethernet
- Loop to a specific node on the Ethernet

The UNA is used to loop messages on the Ethernet circuit. The Ethernet circuit must be in the ON state with service enabled in order to be tested. Use the following commands to enable service and to set the Ethernet circuit device UNA (controller number 0) to the ON state:

```
NCP SET CIRCUIT UNA-0 SERVICE ENABLED
NCP SET CIRCUIT UNA-0 STATE ON
```

A.3.1 Random Node Loopback Testing

You can use the following command to send a test message to all nodes on the Ethernet via a multicast message. The first node to respond is designated as the test node (provided that the COUNT is greater than 1):

```
NCP LOOP CIRCUIT UNA-0
NCP LOOP CIRCUIT UNA-0 COUNT 50
```

A return message is displayed that indicates the node on the Ethernet that responded to the loop:

```
NCP -- Loop Succeeded
       Physical Address=AA-00-04-00-23-04, Node = 35 (NODE )
```

A.3.2 Specific Node Loopback Testing

You can specify the physical address or the node name/address for the circuit on a specific remote node that you want to test.

NOTE

> Nodes on Ethernet circuits are identified by unique Ethernet addresses. An Ethernet address is represented as six pairs of hexadecimal digits separated by hyphens (for example, AA-00-00-03-67-FF).

Each device on the Ethernet has a unique hardware address (in read-only memory) that has been assigned to it by the manufacturer. DECnet sets an Ethernet physical address for each UNA that replaces the hardware address as the address to that the UNA responds. The UNA will continue to respond to the Ethernet physical address unless this address is reset to the hardware address value. (However, the Ethernet physical address reverts back to the hardware address value only when the node is powered OFF or rebooted.)

NOTE

> "Physical address" refers to the address to which the node currently responds. This address could be either the hardware address or the Ethernet physical address set by DECnet.

You can specify the NODE parameter for the device on the remote node that you wish to test. The following command loops messages through the local device UNA-0 to NODE NEPTUN:

```
NCP LOOP CIRCUIT UNA-0 NODE NEPTUN
```

Specifying the node address value (such as 226) for the NODE parameter also results in a successful loopback test. The following command loops messages through the local device UNA-0 to NODE (address value) 226:

```
NCP LOOP CIRCUIT UNA-0 NODE 226
```

You can also specify the PHYSICAL ADDRESS parameter along with its value to test to a remote node. The following command loops messages through the local device UNA-0 to the remote node having physical address AA-00-03-00-FF-08:

```
NCP LOOP CIRCUIT UNA-0 PHYSICAL ADDRESS AA-00-03-00-FF-08
```

If a remote node that you want to loop to is not running DECnet, use the PHYSICAL ADDRESS parameter along with the hardware address value for that node.

A.3.2.1 Loopback Test Error Message — If a loopback test does not succeed, you will receive the following error display:

```
NCP -- Loop failed, line communication error Receiver
Unlooped count=1
```

A.3.3 Loopback Assistance

For Ethernet loopback testing you can use an assistant physical address and an assistant node to aid you in interrogating a device on a remote node. The assistant is useful when you are unable to successfully loop to a remote node, but it is known that some other node on the Ethernet can communicate with that remote node.

You can use the assistant by specifying the appropriate HELP parameter for the type of assistance you need:

- HELP TRANSMIT. Assist in transmitting messages to a remote node.
- HELP RECEIVE. Assist you in receiving messages from a remode node.
- HELP FULL. Assists you in both transmitting messages to and receiving messages from a remote node. This is the full sense, or the default method, of using the assistant.

When you have to use assistance in looping to a remote node on the Ethernet, this indicates the existance of a hardware problem with that remote node. The type of assistance that you use will depend on the nature of the problem on the remote node.

If you specify either the ASSISTANT PHYSICAL ADDRESS or ASSISTANT NODE parameters without specifying the HELP parameter, you will receive full assistance (that is, help in both transmitting and receiving information). The following commands show the use of full assistance.

- In the first example, you request the node with Ethernet physical address AA-00-03-00-AB-FC to assist you in testing the node with Ethernet physical address AA-00-03-00-0E-08:

```
NCP LOOP CIRCUIT UNA-0 PHYSICAL ADDRESS AA-00-03-00-0E-08 -
NCP ASSISTANT PHYSICAL ADDRESS AA-00-03-00-AB-FC
```

- In the second example, you request node MARS to assist you in testing node SATURN:

```
NCP LOOP CIRCUIT UNA-0 NODE SATURN ASSISTANT NODE MARS
```

  You can also specify the ASSISTANT PHYSICAL ADDRESS or ASSISTANT NODE parameters along with the specific HELP parameter for the type of assistance you desire.

- The following command specifies the HELP RECEIVE parameter to request node MARS to assist you in receiving information for node SATURN:

```
NCP LOOP CIRCUIT UNA-0 NODE SATURN ASSISTANT NODE MARS HELP RECEIVE
```

- The following command specifies the HELP TRANSMIT parameter to request the node with (address value 224) to assist you in transmitting information to node SATURN:

```
NCP LOOP CIRCUIT UNA-0 NODE SATURN ASSISTANT NODE 224 HELP TRANSMIT
```

Appendix B
DTS/DTR Test Programs

DTS and DTR are the DECnet-11M/M-Plus transmitter and receiver test programs.

B.1 Types of Tests

There are four basic tests provided by DTS and DTR:

- Connect test
- Data test
- Disconnect test
- Interrupt test

Each test is divided into a set of subtests. The tests and subtests are described in the following sections.

B.1.1 Connect Tests

Connect tests verify the ability of the network software to process connect, connect accept, and connect reject requests with and without optional user data. Connect tests that the user can perform are:

- Connect reject without user data
- Connect accept without user data
- Connect reject with 16 bytes of standard user data
- Connect accept with 16 bytes of standard user data
- Connect reject with received user data used as reject user data
- Connect accept with received user data used as accept user data

B.1.2 Data Tests

Data tests provide a full range of test capabilities from the very simple data sink operation through data integrity checking. Data tests the user can perform are:

- **Sink test.** DTR ignores all data received. No sequence or content validation is performed.
- **Sequence test.** Data messages transmitted by DTS to DTR include a 4-byte sequence number. If a message is received out of sequence, DTR aborts the logical link and the test.
- **Pattern test.** Data messages transmitted to DTR have both a sequence number and a standard data pattern. If either the sequence number or the received data does not match the expected data, DTR aborts the logical link and the test.
- **Echo test.** Data messages received by DTR are transmitted back to DTS. There is no sequence or data validity checking done by either DTR or DTS.

B.1.3 Disconnect Tests

Disconnect tests are designed to determine whether DTS can detect the difference between disconnect and abort sequences generated by DTR as well as receive the proper optional user data. Disconnect tests that can be performed by the user are:

- Disconnect without data
- Abort without user data
- Disconnect with 16 bytes of standard user data
- Abort with 16 bytes of standard user data
- Disconnect with received connect user data used as disconnect user data
- Abort with received connect user data used as abort user data

B.1.4 Interrupt Tests

Interrupt tests provide a full range of test capabilities from very simple data sink operations through data integrity checking. Interrupt tests that the user can perform are:

- **Sink test.** DTR ignores all interrupt data received. No sequence or content validation is performed.
- **Sequence test.** Interrupt messages transmitted by DTS to DTR contain a 4-byte sequence number. If a message is received out of sequence, DTR aborts the logical link and the test.

- **Pattern test.** Interrupt messages transmitted to DTR have both a sequence number and a standard data pattern. If either the sequence number or the data pattern is not received with the expected data, DTR aborts the logical link and the test.
- **Echo test.** Interrupt messages received by DTR are transmitted back to DTS. There is no sequence or data validity checking done by either DTR or DTS.

B.2 Operational Characteristics

DTR functions as a slave to DTS. DTS initiates each test by issuing a connect request to DTR. Parameter information pertinent to the type of test requested is passed by DTS to DTR in the optional data of the connect request. DTS has a user interface that enables the user to specify the test to be performed. Sufficient parameters are available to allow for a variety of tests, including test duration, buffer size, and buffering level.

The version of DTR supplied on the kits supports a maximum of three logical links, thus allowing three concurrent tests. DTS can handle only a single test and logical link, but the user can invoke multiple copies of DTS.

The DTS command syntax allows for two types of buffer level options: BUFS and FLOW. BUFS is used by DTS only and FLOW is used by DTR only. The BUFS parameter specifies the transmit buffering level for DTS. This is the number of transmit requests that DTS attempts to keep outstanding to the network. In an echo test, the BUFS parameter also specifies the receive buffering level for DTS. The parameter of the FLOW option specifies the receive buffering level for DTR.

Both DTS and DTR have 512 bytes of buffer space built into each task. The buffer space is at the end of each task image. To increase the buffer space of either task, the user need only employ the /INC option at installation time. DTS requires at least (BUFSMSG) bytes of buffer space of a data test and ((BUFS+1)MSG) bytes for an echo test. DTR requires (FLOWMSG) bytes of buffer space for each data test.

B.3 DTS Command Syntax

Formats for DTS commands are described below. The following conventions apply to the presentation and use of the DTS commands:

Conventions:

Brackets [ ]	Designate an optional parameter.
UPPERCASE	Designates the actual code that appears in the call.
lowercase italic	Designates a parameter that is replaced with an actual value when assembled.
No space	Do not use a space after the prompt > in a command string.
Default values	Values specified as defaults are valid only for the first test. Before the conclusion of each test, DTS updates the default values with the parameters of the next test, which become the default values for the next test. Thus, it is necessary to specify only those parameters that are to be changed in the next test.
Input	DTS accepts an indirect command file as input.

Format:

DTS>[*nodename*[*aci*]]::[/TEST=*type*][/PRINT=*choice*]

Arguments:

nodename

Name of the node where DTS resides; the default is the local node name. The name must be terminated by a double colon [::] or underscore for Version 2 compatibility.

aci

Access control information, in the following format:

[/*user-id*[/*password*[/*account*]]]

For more information, refer to the *DECnet-RSX System Manager's Guide*.

type

Test type:

CON	Connect test
DIS	Disconnect test
DATA	Data test (default)
INT	Interrupt test

choice

YES (default) or NO to indicate whether DTR messages are to be printed at the remote node as well as at the local node.

NOTE

The print option is used by DTR to either display or suppress message output to the local console device. Messages include both test results and error messages.

B.3.1 Connect Test

Format:

CONNECT TEST>[*type*][/DATA=*data*]

Arguments:

type

Type of connect test:

ACC	Connect accept test
REJ	Connect reject test (default)

data

Type of user data:

NONE	No optional user data (default)
STD	Standard user data
RCVD	Return connect user data

B.3.2 Disconnect Test

Format:

DISCONNECT TEST>[*type*][/DATA=*data*]

Arguments:

type

Type of disconnect test:

DSC	Disconnect test (default)
ABT	Abort test

data

Type of user data:

NONE	No optional user data (default)
STD	Standard user data
RCVD	Return connect user data

B.3.3 Data Test

Format:

DATA TEST>[*type*][/MSG=*mmm*][/BUFS=*bbb*][/TIME=*time*]
[/BAUD=*nnnnnn*][/FLOW=*flow*]

Arguments:

type

Type of data test:

SINK	Sink test (default)
SEQ	Sequence test
PAT	Pattern test
ECHO	Echo test

mmm

Data message length in bytes: *mmm* must be greater than 0 for a sink or echo test, greater than 4 for a sequence test, and greater than 5 for a pattern test. The maximum value is 1024 bytes. The default is the NSP segment size.

bbb

DTS transmit buffering level, in the range of 1 to 16. The default is 1.

time

Test duration:

nnnS	seconds
nnnM	minutes
nnH	hours (maximum is 18)
nnn	defaults to seconds

The default is 2M.

nnnnnn

Line baud rate in bits per second (default is 0). If *nnnnnn* is specified, the percentage of line bandwidth used by data transfers is calculated. This percentage is inaccurate for baud rates exceeding 64K bps.

flow

Flow control type or DTR receive buffering level, in the range of 1 to 16:

NONE	No flow control
SEG:nn	Segment flow control (invalid for RSX)
MES:nn	Message flow control

The default is MES:1. SEG:*nn* is invalid for RSX DTR because the network interface does not provide that capability.

B.3.4 Interrupt Test

Format:

INTERRUPT TEST [*type*][/MSG *mm*][/TIME *time*]

Arguments:

type

Type of interrupt test:

SINK	Sink test (default)
SEQ	Sequence test
PAT	Pattern test
ECHO	Echo test

mm

Interrupt message length in bytes: *mm* must be greater than 0 for a sink or echo test, greater than 4 for a sequence test, and greater than 5 for a pattern test. The maximum value (and the default) is 16 bytes.

time

Test duration:

nnnS	seconds
nnnM	minutes
nnH	hours (maximum is 18)
nnn	defaults to seconds

The default is 2M.

Index

N

O

P

Q

R

S

T

U

V

W

X

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) ______

Name ______ Date ______

Organization ______

Street ______

City ______ State ______ Zip Code or Country ______

Do Not Tear - Fold Here and Tape

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 33 MAYNARD MASS

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE DOCUMENTATION
1925 ANDOVER STREET TW/E07
TEWKSBURY, MASSACHUSETTS 01876

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line